정보시스템

성능관리 지침

Guideline
for Performance Management
of Information System

국무조정실 정보통신부

정보시스템 성능관리 지침

Guideline for Performance Management of Information System

1. 정보시스템 운영관리 지침 개요서
2. 정보시스템 운영관리 지침

관리요소별 세부 지침

관리요소별 세부 지침	해당 10대 관리요소
3. 정보시스템 구성 및 변경관리 지침	구성 및 변경관리
4. 정보시스템 운영상태관리 지침	운영상태관리
5. 정보시스템 성능관리 지침	성능관리
6. 정보시스템 장애관리 지침	장애관리
7. 정보시스템 재해복구 지침	백업관리
8. 정보시스템 백업 지침	
9. 서비스데스크 운영관리 지침	사용자지원관리
10. 전산실 관리 지침	전산실관리
11. 정보시스템 운영 아웃소싱 관리 지침	운영아웃소싱관리
12. SLA를 강화한 정보시스템 운영계약 참조모델	

2005. 12

제·개정 이력표

버전	제·개정 내용	일자
V1.0	정보시스템 성능관리 지침 제정	2005. 12

Contents

표차례

그림차례

1 개요

성능관리는 통합된 정보시스템의 모든 구성요소(서버, 네트워크, DBMS, 응용 소프트웨어)의 효율적인 활동능력을 부여하고, 성능에 관계된 모든 상태를 감시하여, 최적의 서비스 품질과 정보시스템 자원의 효율성을 유지 및 제고시키는 제반 활동이다. 이 같은 성능관리 업무는 최적의 용량을 적시에 확보하기 위한 용량계획의 시점을 제공하고 성능 관련 문제를 사전에 적극적으로 예방(Pro-Active)함으로써, 사용자의 시스템 활용도 및 만족도를 향상시킬 수 있다.

오늘날과 같이 급변하는 비즈니스 환경에서 고객에게 제공되어지는 정보 서비스의 양과 질과 시점이 비즈니스의 성패를 좌우한다 해도 과언이 아닐 것이다. 그러나 이제는 모든 조직이 서비스의 질을 통해서만 고객을 만족시킬 수 있다는 점과 그것을 결정하는 핵심 부분이 바로 성능(Performance)임을 깊이 인식하여야 한다. 공공기관 정보화 시스템에 대한 질적 향상은 최근 IT 인프라스트럭처 관리의 주된 트렌드(Trend)인 RTE(Real Time Enterprise)와 RTI(Real Time Infrastructure)의 배경을 통하여 성능관리에 대한 필요성을 이해할 수 있다. RTE는 '최신(실시간) 정보를 기반으로 하여, 핵심적인 프로세스를 관리하고 수행할 때 발생할 수 있는 지체 현상을 지속적으로 제거함으로써, 경쟁력을 극대화하는 경영체제(Gartner, 2002)' 라고 정의하며, 실시간 모니터링(Monitoring), 실시간 분석(Analysis), 실시간 실행(Execution)을 전제한다. RTI는 Business Policies와 SLA(service Level Agreements)를 기반으로 동적(자동)으로 최적화하여 비용은 줄이되 서비스의 민첩성과 품질을 향상시킬 수 있도록 구현된 IT 인프라스트럭처를 의미한다. 이러한 RTE/RTI 관리 체계에서 핵심 이슈는 말 그대로, '실시간(Real Time)' 이라는 키워드이며, 따라서 그 이면에 최적의 성능관리(Performance Management)가 지원되어야 한다.

정보시스템의 성능관리는 전반적으로 다음과 같은 프로세스를 필요로 한다. 첫째는 정보시스템 각각의 구성요소(서버, 네트워크, DBMS, 응용 소프트웨어)에 대한 성능 및 상태를 측정(Monitoring)하는 과정이다. 이 과정에서는 정보시스템 성능의 최적화를 위하여 다음 단계에서 이용될 성능 데이터를 수집한다. 둘째는 위에서 추출된 자료를 기초로 정보시스템의 성능과 상태를 분석하는 과정이다. 이 과정에서는 구성요소에 대한 개별적인 분석과 전체적인 통합 성능 분석 환경이 마련되어야 한다. 셋째는 위의 두 과정에서 얻어진 자료를 기초로 성능 제고를 위하여 시스템 구성요소를 조정(Tuning)하는 과정과 검증(Feedback)하는 과정이다. 이것은 정보시스템 전

체의 성능 향상을 위한 작업이므로 각 구성요소별 연계분야와의 상호관련성을 고려하여 진행하여야 한다.

본 지침은 3장에서 성능관리에 대한 전반적인 대상 및 범위에 대하여 상세하게 기술한다. 성능관리 대상을 서버, 네트워크, DBMS, 응용 소프트웨어로 나누어 특성과 아키텍쳐를 살펴보고 정보시스템 운영관리의 연계분야들과의 고려사항을 설명한다. 4장에서는 성능관리 프로세스의 사전준비로서 계획 수립 및 요구사항 검토, 성능 측정 및 분석 방안 수립, 성능 측정 환경 구현에 대하여 설명하고 있으며, 각 정보시스템 구성요소별 데이터 수집 및 분석, 성능 개선 방안 수립, 성능 개선 실행 및 검증, 통합 성능관리에 대하여 구체적인 지침을 제시하고 있다.

정보시스템의 성능관리를 위해 필요한 업무 절차 및 산출물, 구체적인 문서양식 등은 업무를 수행하는 기관의 규모, 업무 분야, 시스템의 종류 및 유형에 따라 많은 차이가 있을 수 있으므로, 하나의 정보시스템 성능관리 지침을 모든 기관에 동일하게 적용하는 것은 현실적으로 불가능하다. 따라서 기관의 형편에 맞도록 본 지침을 적절하게 조정(customize)하여 기관별 성능관리 지침서 및 절차서를 반드시 작성하고, 이를 업무에 활용해야 한다. 기관별 지침서 및 절차서에 대한 설명은 '3장 대상 및 범위'와 '4장 성능관리 프로세스' 내용을 참고하여 구체적으로 작성, 실무에 적용되어야 한다.

본 지침의 활용을 통해, 각 기관은 정보시스템의 성능관리를 위한 종합적인 체계를 마련할 수 있으며, 제공되는 운영 서비스에 대한 정확성과 신뢰성을 증진시키고, 성능관리자의 관리 능력 향상을 도모할 수 있다.

2 지침의 구성 및 범위

본 지침은 정보시스템의 성능을 최적으로 유지, 운영하기 위하여 현재 및 과거(성능 추이 정보)의 시스템 성능 상황을 분석하여 이를 근거로 성능조정(튜닝)을 함으로써 정보시스템을 안정적이고 효율적으로 운영하기 위한 방법 등에 대한 내용을 주로 다루고자 한다.

3장에서는 정보시스템 성능관리의 개념 및 성능관리의 대상, 범위, 역할 등에 대하여 전반적으로 살펴보고 또한 성능관리와의 연계분야 및 본 지침에서 전반적으로 사용되고 있는 용어의 의미를 정리한다.

4장에서는 정보시스템 성능관리를 하기 위한 분야별 사전 준비사항 및 성능데이터 수집, 분석 방법, 성능 개선 대상 선정 및 성능 조정(튜닝), 개선효과 검증 등에 대하여 기술한다.

부록에서는 본 성능관리 지침에서 제시하고 있는 각 프로세스에 해당하는 성능관리를 위한 각종 양식들을 첨부한다.

본 지침에서 다루는 정보시스템의 범위는 서버, 네트워크, 데이터베이스, 응용 소프트웨어로 한정한다. 또한 본 지침에서 다루고자 하는 4가지의 정보시스템 각 분야를 총 망라한 통합 성능관리를 위한 통합적 성능관리 분야도 추가로 기술한다.

각 분야별 성능관리 및 통합적 성능관리 시스템 구축을 위해서는 해당 성능관리 기능을 제공하는 자동화된 성능관리 도구가 일부 필요하나 본 지침에서는 관련된 특정 솔루션을 지정하여 언급하지는 않았다. 따라서 성능관리 도구의 도입이 필요시 각 기관은 성능관리 대상 및 IT 환경에 맞는 제품을 선정하여 도입, 적용하면 될 것이다.

3 성능관리 개요

3 1 성능관리 개념

3 1.1 정의

최초의 컴퓨터가 발명된 이후 정보시스템은 1970년대까지는 제한된 업무 영역 및 소수의 IT 전문가에 의해서만 활용되었으나 PC가 널리 보급되기 시작한 1980년대 이후부터는 일반 사용자도 전산실이 아닌 사무실내에서 정보시스템을 활용하기 시작하였다. 하지만, 이 때는 주로 문서작성이나 스프레드쉬트를 위한 단독형(stand-alone) 프로그램만을 사용하였기 때문에 시스템의 성능저하나 장애는 해당 사용자에게만 영향을 주었다.

1990년대 이후 네트워크 컴퓨팅이 본격화되기 시작하면서 PC, 워크스테이션, 중대형 컴퓨터 등 다양한 정보시스템 자원들이 하나의 정보시스템으로 연결되기 시작하였고, 단순한 업무 처리를 지원하는 수준을 벗어나 보다 복잡하고 방대한 자료를 효율적으로 처리할 수 있는 전략적 도구로 인식되기 시작하였다. 이를 기점으로 정보시스템의 적용 범위와 기능은 갈수록 고도화되고 다양화 되었으며, 더불어 이를 지원하기 위한 IT 기술과 솔루션이 더욱 복잡하고 다양하게 적용되었고 다수의 기업 또는 기관들은 조직 내부의 사용자는 물론 전 세계인을 대상으로까지 정보서비스를 제공하는 등 정보시스템을 둘러싼 내·외부 환경들이 급속하게 변화하고 있다.

과거에는 소수의 정보시스템 자원들에 대하여 제한된(업무) 시간 동안 제한된 기준들에 대해서만 관리하여도 충분하였으나 현재는 다수의 자원들에 대하여 일일 24시간 동안 다양한 기준들에 대하여 관리하지 않으면 안되는 상황이 되었고, 단지 몇 분이라고 하더라도 목표응답시간 내에 업무가 처리되지 않거나 사용이 불가능하게 되면 해당 정보시스템을 이용하는 사용자에게 엄청난 불편을 주는 것은 물론이고 대외적인 이미지 손상 및 막대한 경제적 손실을 끼치는 경우를 흔치 않게 볼 수 있다.

따라서, 정보시스템이 서비스를 제공하도록 정의된 시간 동안 목표응답시간 내에 모든(정의된 부하량 내의) 사용자의 요청을 성공적으로 처리할 수 있도록 유지, 관리 및 개선하는 활동이 더욱 중요하게 되었으며, 이를 정보시스템의 성능관리라고 한다.

3 1.2 성능 지표

정보시스템의 서비스 품질(QoS)을 결정하는 속성들 중의 하나인 성능을 나타내는 일반적인 지표에는 다음과 같은 것들이 있다〈표 3-1〉.

〈표 3-1〉 성능을 나타내는 일반적인 지표

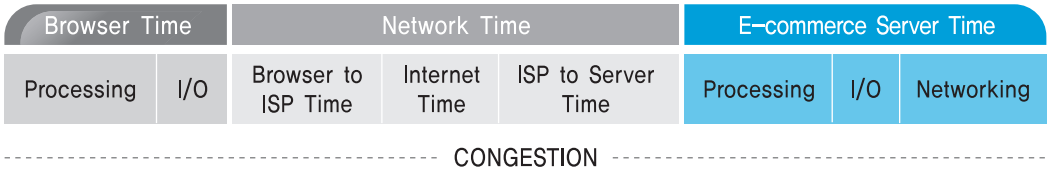
성능 지표	정의	단위(예시)	목표
응답 시간 (Response Time)	작업 처리를 요청한 시간으로부터 이를 시스템이 처리하여 결과를 보여줄 때까지 소요된 시간	초	낮춤
시간당 처리량 (Throughput)	시스템이 성공적으로 처리한 단위 시간당 요청(트랜잭션) 처리 건수	TPS[주1] OPS[주2]	높임
자원 사용량 (Utilization)	자원(CPU, 메모리 등)들의 용량 중 실제 사용하고 있는 값의 비율	%	높임
효율성 (Efficiency)	시간당 처리량을 자원사용량 또는 비용으로 나눈 값	% tpmC[주3]	높임

※ 주1 : TPS (Transactions per Second: 초당 트랜잭션 처리건수)
주2 : OPS (Operations per Second: 초당 요청 처리건수)
주3 : tpmC (Transactions per Minute per Cost: 단위 비용당 분당 처리건수)

이 중 가장 중요한 성능 지표인 응답 시간과 시간당 처리량에 대하여 좀 더 살펴보기로 한다.

가. 응답 시간

시스템 사용자(End User) 관점에서 보았을 때 일반적인 웹(Web) 정보시스템의 경우 응답 시간의 구조는 (그림 3-1)과 같다.



(그림 3-1) 응답 시간 구조

이 과정을 자세히 살펴보면, 응답 시간 – 정확하게는 사용자 응답 시간(End User Response Time) – 은 크게 클라이언트 시간, 네트워크 시간, 서버 시간으로 이루어진다.

클라이언트 시간은 HTTP(HTTPS) 요청을 서버로 전송하고 서버로부터의 결과를 브라우저에 표시하기 위한 I/O(Input/Output) 및 CPU 처리시간이 포함된다. 네트워크 시간은 HTTP(HTTPS) 요청이 브라우저로부터 인터넷 서비스 업체(ISP, 예: 한국 통신, 하나로 통신 등)까지 전송되는데 걸리는 시간, 인터넷 서비스 업체로부터 인터넷을 거쳐 웹 사이트의 서버까지 전달되는데 걸리는 시간, 처리 결과가 반대의 경로로 브라우저에 전달되는데 걸리는 시간으로 이루어진다. 마지막으로, 서버 시간은 웹 사이트에서 HTTP(HTTPS) 요청을 처리하기 위해 서버에서 소요되는 I/O 및 CPU의 처리시간과 내부 서버들간의 네트워크 통신 시간을 포함한다.

이 3가지 시간은 시스템의 자원들(CPU, 디스크, 네트워크)을 사용하기 위해 대기하는 시간인 경합 시간(congestion time)이 포함되어 있다. 경합 시간은 시스템이 동시에 처리해야 하는 요청 건수에 따라 달라지는데 동시 요청이 많으면 많을수록 이 경합 시간도 늘어나게 된다.

시스템의 관점에서 보았을 경우에도 응답 시간은 하나의 소프트웨어 또는 하드웨어 모듈이 타 모듈에서 제공하는 서비스를 요청한 후 응답을 받을 때까지의 소요 시간으로 정의할 수 있으므로 위 (그림 3-1)과 같이 「사용자 응답시간 = 브라우저 응답시간 + 네트워크 응답시간 + 서버 응답시간」이 됨을 알 수 있다.

나. 시간당 처리량

시간당 처리량은 일반적으로 〈표 3-2〉와 같은 방법으로 측정되는데, 이를 측정할 때는 요청(트랜잭션)의 성격을 명확히 이해하여야 한다. 예를 들어, 온라인 처리(OLTP : Online Transaction Processing) 시스템의 경우 시간당 처리량은 초당 트랜잭션 처리 건수(TPS : Transactions Per Second)로 측정되지만, 트랜잭션은 처리 내용과 자원 사용량에 따라 판이하게 다르기 때문에 무엇을 하나의 트랜잭션으로 보는지를 명확히 규정하지 않으면 아무런 의미 없는 수치가 되어버린다.

TPC(Transaction Processing Performance Council)는 온라인 처리 시스템에 적용되는 TPC-C 벤치마크를 제공하고 있는데, 이는 전형적인 주문 처리 시스템의 트랜잭션을 정의하여 각 제품별로 1분당 트랜잭션 처리량 및 트랜잭션당 소요 비용(tpmC)을 측정하는 것이다.

웹 시스템의 경우에는 SPEC(Standard Performance Evaluation Corporation)에서 제공하는 Web96, Web99, Web2005 등의 벤치마크를 활용할 수 있는데, Web96은 정적인(static) 웹페이지만을 대상으로 하였으나 Web99 이후 동적인(dynamic) 컨텐츠 및 HTTP 1.1 Persistent Connection 등을 워크로드 모델에 포함하여 실 환경에 근접한 측정환경을 구현하고 있다.

OPS(Operations per Second)는 Web96의 측정단위였으나, Web99 이후 최대 동시 연결 개수가 주요 측정지표로 변경되었으며(OPS는 보조 측정 단위) 웹 정보시스템의 경우 현재는 OPS보다는 초당 페이지 처리 건수가 보다 중요한 개념으로 사용되고 있다.

한 가지 유의해야할 점은 정보시스템의 용량 또는 성능을 나타내기 위한 값으로 시간당 처리량을 사용할 경우에는 최대값을 사용해야 한다. 즉, 시간당 처리량은 정보시스템에 가해지는 부하량에 좌우되며 운영 중에는 최대값을 측정하기 어려우므로 일반적으로는 부하 테스트 등을 통해 산출한다.

시간당 처리량(throughput) = MIN(최대 처리량, 작업 부하)

〈표 3-2〉 시간당 처리량 매트릭스

시스템	측정 방법
온라인 트랜잭션 처리(OLTP) 시스템	초당 트랜잭션 처리 건수(Transactions Per Second : TPS)
	분당 트랜잭션 처리 건수 비용(TPC-C : tpmC)
웹 사이트	초당 요청 처리 건수(HTTP requests/sec)
	초당 페이지 처리 건수(Page Views per Second)
전자상거래 사이트	초당 웹 처리 건수(TPC-W : WIPS(Web Interactions Per Second))
	초당 세션 수(Sessions per Second)
라우터	초당 패킷 처리 건수(Packets per Second : PPS)
	초당 데이터 전송량(MB transferred per Second)
CPU	초당 명령 수행 건수(Millions of Instructions per Second : MIPS)
	초당 부동 소수점 계산 수행 건수(Floating Point Operations per Second : FLOPS)
디스크	초당 입출력 건수(I/Os per Second)
	초당 데이터 전송량(KB transferred per Second)
E-mail 서버	초당 메시지 전송 건수(Messages Sent Per Second)

다. 성능 지표간의 관계

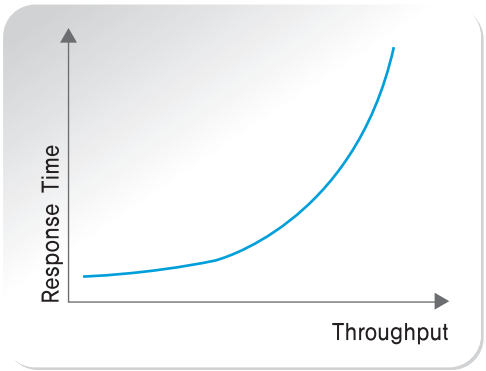
일반적으로 시간당 처리량이 큰 시스템의 경우 응답 시간이 작은 것이 보통이지만 반드시 그런 것은 아니다. 예를 들어, 단일 노드로 구성된 시스템을 처리 업무를 분리하여 이중화 구성하는 경우 전체적인 처리 용량이 증가되어 시간당 처리량은 증가하게 되지만 이중화 노드간의 동기화

등의 오버헤드로 인해 하나의 요청에 대해서는 다소의 응답 시간 지연이 발생하게 된다.

따라서, 응답 시간과 시간당 처리량은 별개의 개념으로 인식하여야 하지만 동일 시스템 내에서는 두 지표 간에 (그림 3-2)와 같은 일정한 관계를 가지게 된다.

$$\text{시간당처리량(throughput)} = \frac{\text{동시사용자수}}{\text{평균응답시간+Think Time}}$$

여기서 Think Time은 사용자가 응답을 받은 후 다음 요청을 보낼 때까지의 소요시간을 의미하며, 동시 사용자는 특정 시점을 기준으로 해당 시점에 서비스를 요청하였거나 이전의 요청에 대한 응답을 받은 후 다음 요청을 보낼(예: 10분 이내) 사용자들을 의미한다. 따라서, 현재 시점에 시스템에 접속되어 있는 사용자 또는 요청을 보내고 응답을 기다리는 사용자(동시 요청자)와는 다른 개념임을 분명히 인식하여야 한다.



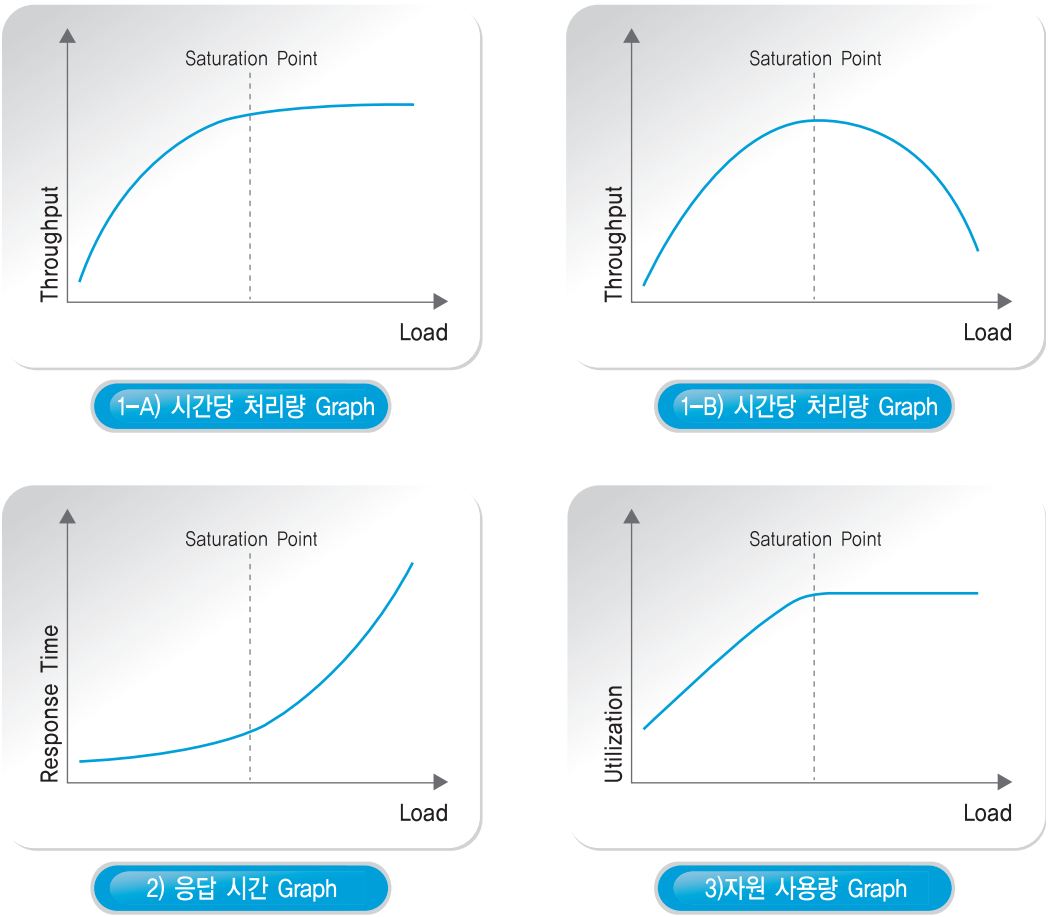
(그림 3-2) 시간당 처리량, 응답 시간 그래프

라. 부하량과 성능 지표와의 관계

정상적인 시스템의 경우 부하량에 따른 응답 시간과 시간당 처리량의 변화는 다음의 (그림 3-3)과 같으며, 'Saturation Point'는 응답 시간이 급격히 증가하기 시작하고 시간당 처리량이 거의

일정하게 유지되는 지점을 의미한다. 이 때의 부하량이 시스템에서 지원 가능한 최대 부하량이다.

(그림 3-3)에서 1-B와 같이 시간당 처리량이 Saturation Point 이후 일정하게 유지되는 것이 아니라 오히려 감소(Thrashing)하는 경우도 있는데 대부분은 시스템의 자원 부족으로 인해 발생하게 된다.

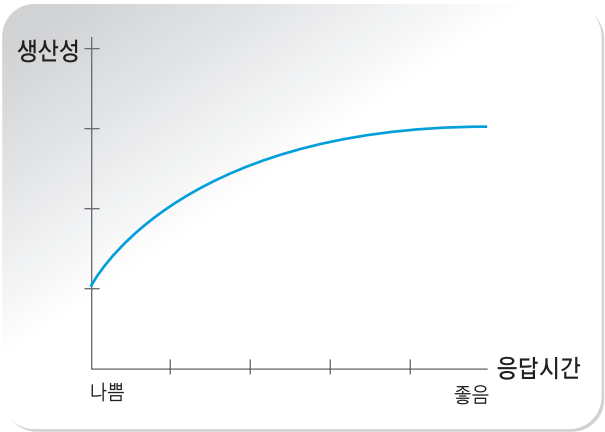


(그림 3-3) 전형적인 시간당 처리량, 응답 시간 및 자원 사용량 그래프

3 1.3 성능관리 활동

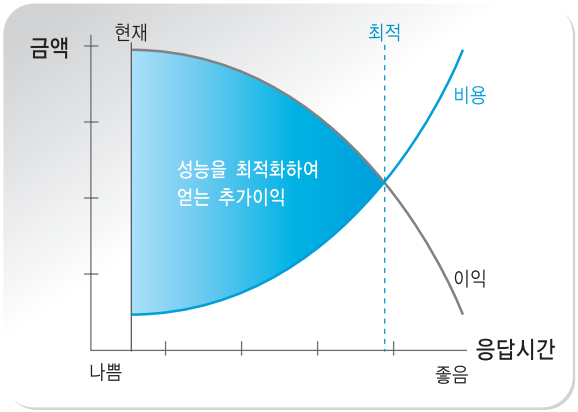
성능관리는 정보시스템을 구성하는 서버, 데이터베이스, 어플리케이션, 네트워크 등의 자원에 대한 운영상태관리 데이터를 수집하여 세부 자원(CPU, 메모리, 디스크 I/O 등)들 및 구성요소(어플리케이션, 데이터베이스 등)들의 운영상태와 운영조건(사용자 접근 환경, 부하량 등) 등을 종합적으로 분석하고 개선이 필요할 경우 정의된 목표응답시간(또는 목표처리량)을 만족시키는데 가장 효과적인 영역 및 구체적인 실행 계획들을 도출하여 이를 수행하고 그 결과를 검증하는 활동을 포함한다.

이를 위해 가장 중요한 부분이 성능 목표를 수립하는 것으로 이를 위해서는 시스템 및 조직의 목표, 투자비용 대비 효과를 고려하여야 한다. (그림 3-4)는 정보시스템의 성능과 업무 처리의 생산성간의 관계를 나타낸다.



(그림 3-4) 응답 시간, 생산성 그래프

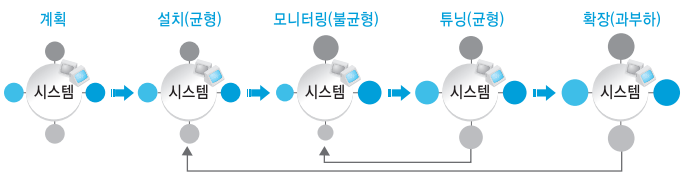
(그림 3-4)에서 보듯이, 정보시스템의 응답 시간을 개선하는데 따라 생산성이 선형으로 증가하지는 않으므로 최적의 성능 목표를 정의하는 것이 중요하며, 다음 (그림 3-5)와 같은 방법으로 이를 결정할 수 있다.



(그림 3-5) 최적의 서비스 레벨 결정

또한, 시스템 파라미터 변경, 어플리케이션 및 데이터베이스 튜닝 등이 적합하지 않거나 효과가 없을 경우, 또는 조직의 신규 업무 요구사항에 의해 기 정의된 부하량(Workload Model)을 장기적으로 초과할 것으로 예측되는 경우에는 H/W, S/W 자원에 대한 증설 및 증설 결과 예측(시뮬레이션) 등을 포함하는 용량계획 수립 활동도 넓은 의미에서 성능관리에 포함된다고 볼 수 있다.

(그림 3-6)은 정보시스템의 도입에서부터 구축, 운영 및 진화의 단계를 거치는 동안 수행되는 성능관리의 개념을 나타내는 것으로 설치단계에서는 모든 자원이 균형을 맞추어 운영되지만 구축단계 또는 운영단계에서는 어플리케이션, 하드웨어/소프트웨어 구성 및 사용량의 변경 또는 증가로 인해 시스템 자원간의 불균형이 발생하게 된다. 해당 정보시스템의 서비스 수준에 정의된 목표치를 넘어섰거나 넘어설 것으로 예상되는 경우 성능 조정(튜닝) 활동을 통해 자원들간의 균형을 맞추거나 자원 증설 등의 정보시스템 확장을 통해 설치단계와 같은 자원들간의 균형을 맞추게 된다.

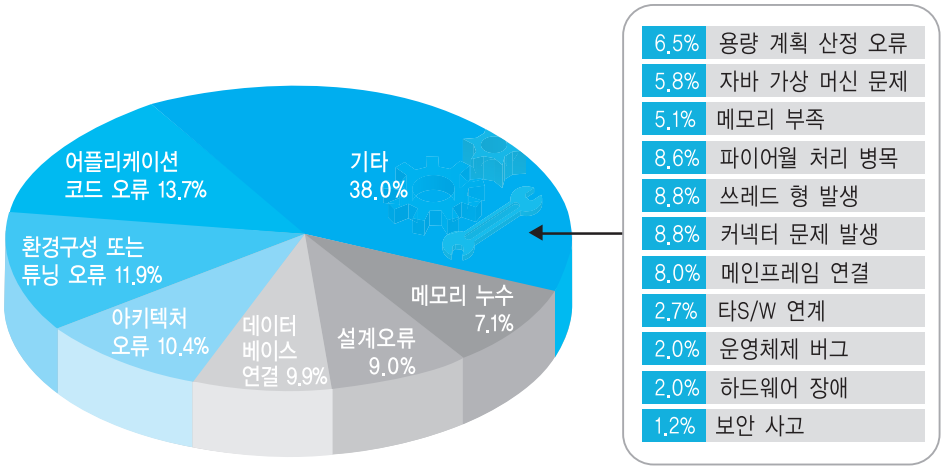


(그림 3-6) 성능관리 개념도

이러한 유지, 관리 및 개선 활동은 단지 성능상의 문제점이 발견되어 이를 해결하는 활동으로서 수행되는 것이 아니라 정보시스템이 운영되는 동안 지속적으로 수행하도록 하여 사전에 문제가 발생되지 않도록 예방하는 활동이 더욱 중요하며 이를 통해 궁극적으로 업무 처리의 효율성을 확보할 수 있는 기반을 제공할 수 있다.

3 1.4 성능 저하 요인

최근에 발생하고 있는 정보시스템의 성능저하 요인은 (그림 3-7)과 같다.



(그림 3-7) 성능저하 요인

(그림 3-7)에서 알 수 있듯이 정보시스템의 설계 및 개발단계의 오류로 인한 성능 저하 문제가 전체의 약 33%(설계 오류, 아키텍처 오류, 어플리케이션 코드 오류)를 차지하고 있으며 특히, 설계 또는 아키텍처의 오류는 개선에 따르는 비용과 시간이 타 부분에 비하여 막대하므로 정보시스템 구축 시 프로젝트 전 단계에 걸쳐 지속적으로 성능관리를 수행하고 그 결과를 검증하는 것이 중요하다.

3 2 대상 및 범위

3 2.1 서버

가. 개요

서버 성능관리는 정보시스템 운영의 대상이 되는 서버들에 대하여 미리 합의된 서비스 수준에 도달하고 유지하기 위한 성능을 보장하기 위하여 시스템 자원의 사용량 및 응답속도에 대한 기준치(Baseline Measurement)를 설정해 놓고, 이러한 자원의 사용과 관련된 정보들을 주기적으로 수집하여 서비스 수준의 위반사항 및 병목현상을 분석하고, 이를 개선하는 절차를 지속적으로 수행하는 것을 말한다.

또한 자원의 부족 또는 자원의 비효율적인 사용으로 인해 야기되는 문제들에 대해서 즉각적으로 대응하고, 자원사용의 효율성을 제고하기 위하여 균형적인 자원의 재배치와 시스템 환경 및 처리량 변화에 의한 사전 발생가능한 성능문제점을 예측하여 장애상황이 발생하기 이전에 사전조치가 가능하도록 하는 활동을 수행한다.

(1) 서버 성능관리의 목적

서버 성능관리 업무는 최적의 용량을 적시에 확보하기 위한 용량계획의 시점을 제공하고 성능 관련 문제를 사전에 예방함으로써, 사용자의 시스템 활용도 및 만족도를 향상시키기 위하여 수행된다.

(2) 서버 성능관리의 주요 활동

- 시스템 성능 분석 및 튜닝을 할 수 있는 절차를 수립
- 서버자원의 현재 상태 및 기존 하드웨어 자원에 대한 환경 평가
- 시스템 성능 저하를 사전에 발견할 수 있는 임계치 설정 및 활동을 정의

- 현재 또는 미래의 잠재적인 성능 지연을 야기하는 근본 원인의 파악 및 조치, 보고
- 필요한 운영체제의 조정 또는 하드웨어의 변경 작업 요청 등 개선 사항 도출
- 수집된 성능데이터의 경향 분석 및 용량계획을 위한 정보 제공

나. 성능관리 대상 및 범위

서버 성능관리의 대상은 정보시스템 운영관리의 대상이 되는 모든 서버 자원을 포함한다. 중요한 서비스와 낮은 성능을 보이거나, 주기적인 성능 관련 분석이 필요한 시스템 및 해당 시스템의 구성요소를 대상으로 하며, 성능 분석 대상으로 선정된 서버의 구성 요소에 대한 업무의 중요도 및 운영 환경을 고려하여 성능관리의 적용 범위가 조정되어질 수 있다.

성능 모니터링을 위해 각각의 서버와 서버의 구성요소들에 대해 성능 측정이 가능한 환경을 구성한다. 서버의 구성요소에 대한 측정을 위해 OS에서 기본으로 제공하는 명령어를 이용하거나 시스템 성능 측정 도구를 설치하여 자원의 사용현황 및 응답속도, 병목현상에 대하여 데이터를 수집한다.

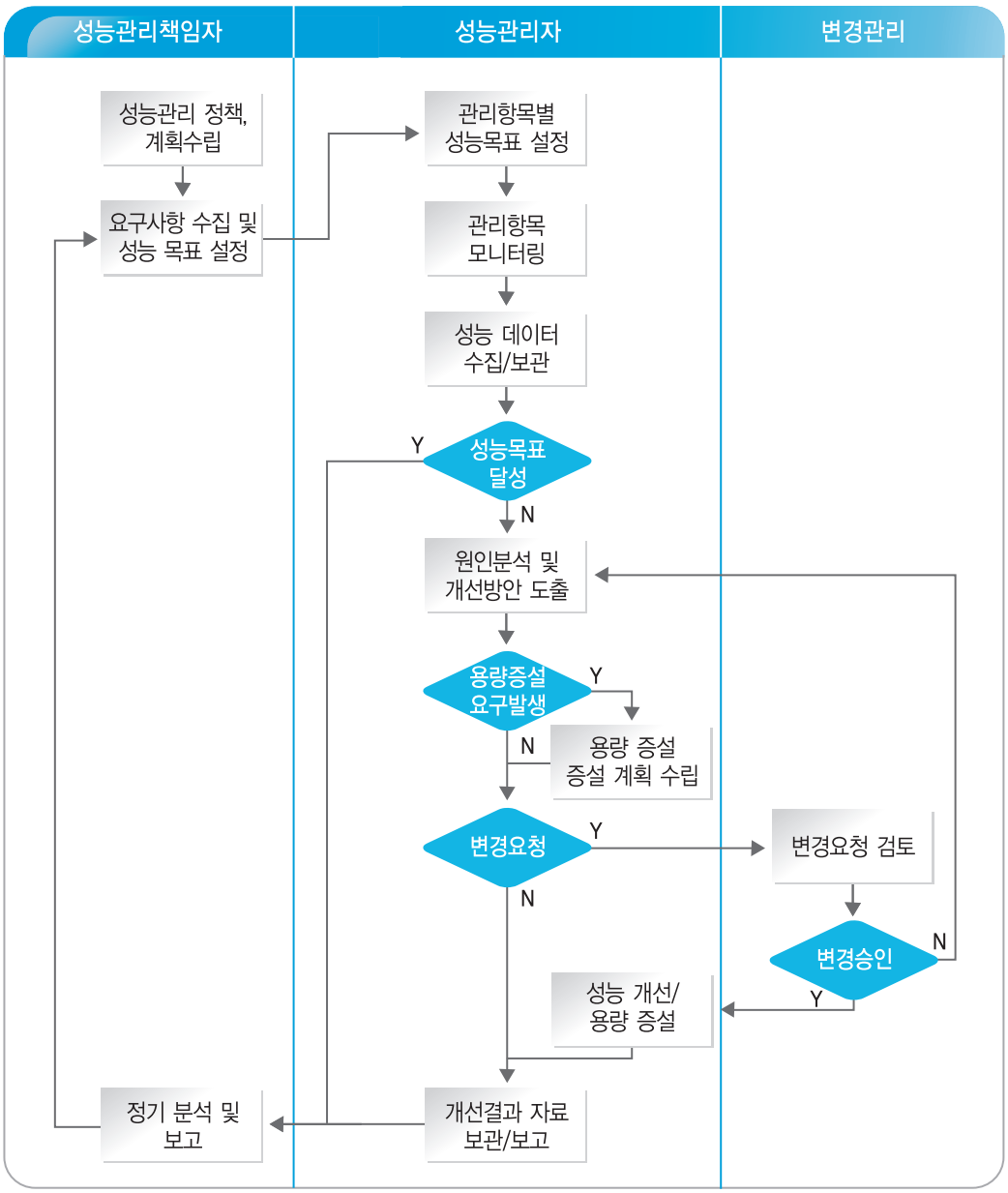
종합적인 성능 분석을 통해 CPU 및 메모리 사용, 스왑(Swap) 비율, 디스크 I/O 비율 및 대기시간 등을 포함한 시스템의 데이터 영역을 다각도로 측정하여야 한다. 이 데이터를 종합적으로 분석하여 현재의 하드웨어 구성으로 서버 성능을 향상시킬 수 있는 방안을 모색하고 시스템 자원을 추가해야 하는 부분을 파악한다. 분석을 통해 작업 유형에 따른 자원 소비량과 전체적인 자원 로딩 비율을 파악 할 수 있다. 서버의 성능 분석은 고성능 IT환경을 위한 첫 번째 수행 단계이며, 보다 정밀한 용량계획 분석을 위한 사전 작업이다.

서버의 주요 성능관리 구성요소는 〈표 3-3〉과 같다.

〈표 3-3〉 서버의 주요 성능관리 구성요소

구성요소	내용
CPU	총 CPU사용율, 시스템 모드 사용율, 사용자 모드 사용율, Run Queue, Pri Queue, 사용자수 등
메모리	총 메모리 사용율, 시스템 및 버퍼 캐쉬, Page In/Out, Swap 공간 사용율 등
디스크	Disk 사용율, Disk I/O Busy, Disk Queue
프로세스	CPU를 집중적으로 사용하는 프로세스, Zombie 프로세스
커널	커널 파라미터 설정을 통한 자원의 적절한 분배
파일시스템	파일시스템 IO Rate, 파일시스템 공간 사용율
네트워크I/O	In 패킷율, Out 패킷율, Collision율, Error율

다. 성능관리 절차도



(그림 3-8) 서버의 성능관리 절차도

3 2.2 네트워크

가. 개요

오늘날의 네트워크 환경은 다기능, 멀티 벤더(Multi-Vendor)의 분산 환경으로 장비 및 프로토콜의 다양성, 네트워크 구조의 복잡성, 지역적 분산 등 광대하고 복잡한 네트워크로 구성되어 있다. 따라서 복잡한 네트워크 구성에 부합하는 신속한 성능, 장애 파악이 어려운 실정이며 사용자들이 요구하는 효율적인 관리의 필요성은 더욱 증가되어 가고 있다.

네트워크 관리란 전산 네트워크가 지속적이고 효율적으로 광범위한 지역에서의 정보 교환, 자원 공유, 치명적인 고장의 대체 기능, 유연성 있는 작업 환경 제공과 같은 목적했던 기능을 수행하고 보다 향상된 서비스를 제공할 수 있도록 전산 네트워크에 연결된 장비와 호스트간의 트래픽에 대한 모니터링을 통해 서비스의 중단 없이 효율적으로 통신 네트워크를 운용할 수 있도록 네트워크 자원의 감시 및 보고와 필요한 경우 제어를 수행하는 제반 활동을 의미한다. 우리가 흔히 네트워크 관리 시스템이라고 부르는 NMS(Network Management System)는 크게 구성, 장애, 성능, 보안, 계정 관리 등 5개 영역으로 구분된다.

- 구성관리 : 상호 연결 서비스의 계속적인 운영을 위한 기능으로 관리 네트워크로부터 네트워크 구성정보를 수집하여, 그 정보를 바탕으로 네트워크 장치의 구성정보를 추가하고 저장하며, 갱신하고 그 구성 형태에 대한 보고를 수행하는 기능

- 장애관리 : 네트워크 자원의 비정상적인 수행에 대한 검출, 분리 및 수정을 수행하는 관리 기능

- 성능관리 : 관리되는 네트워크의 하드웨어, 소프트웨어, 매체의 성능을 평가하여 사용자가 효율적으로 이용 가능하도록 지원하는 기능

- 보안관리 : 네트워크에 연결된 장비에서 발견되는 보안이 필요한 정보에 대한 접근점을 제어함으로써 그 정보를 보호하는 기능

▶ 계정관리 : 네트워크 자원 이용률에 대한 데이터를 수집하여 사용량을 계산하여 그 이용량에 따른 요구 부하 기능

본 지침에서는 상기 5개 영역 중에서 네트워크 성능관리 위주로 설명을 하며, 공공부문에서 운영하는 네트워크 회선 및 장비의 성능관리에 대한 제반 사항을 정하여 안정적이고 효율적인 네트워크 운영/성능관리를 목적으로 한다.

나. 성능관리 대상 및 범위

성능관리는 네트워크 장비 및 관련 링크를 지속적으로 모니터링 하여 이용률, 에러율 등의 성능 지표를 계산하고 사용자들에게 일정 수준의 서비스를 지속적으로 제공하기 위한 기능을 제공하는 것으로 네트워크의 혼잡과 접근 불가 횟수를 줄이는 이점을 갖는다. 또한 성능관리는 그 이용률을 지속적으로 모니터링 하여 링크의 확장들에 관련된 용량 계획을 지원하며 시간대별 네트워크 peak를 파악함으로써 대규모 데이터의 전송 등을 파악하여 좀더 효율적으로 네트워크를 관리하도록 지원할 수 있다.

성능관리는 실시간과 누적 분석 기능의 두 가지 영역으로 나누어 네트워크 성능을 모니터링 한다. 실시간 성능 모니터링은 현재 성능관리 지표(이용률, 에러율 등)의 변화 추이를 분석하는 것이며, 누적 분석 기능은 주기적으로 네트워크 성능 지표를 수집하고 일정 기간(일주일, 한달, 일년)에 대한 성능 통계를 분석하여 그 정보를 장기적인 네트워크 관리에 사용하는 것이다. 일반적인 성능관리는 성능관리 지표(이용률, 에러율)가 임계치를 초과하는지를 평가하고, 어느 시간에 트래픽이 최대 또는 최소가 되는지를 평가하는 방식으로 이루어진다.

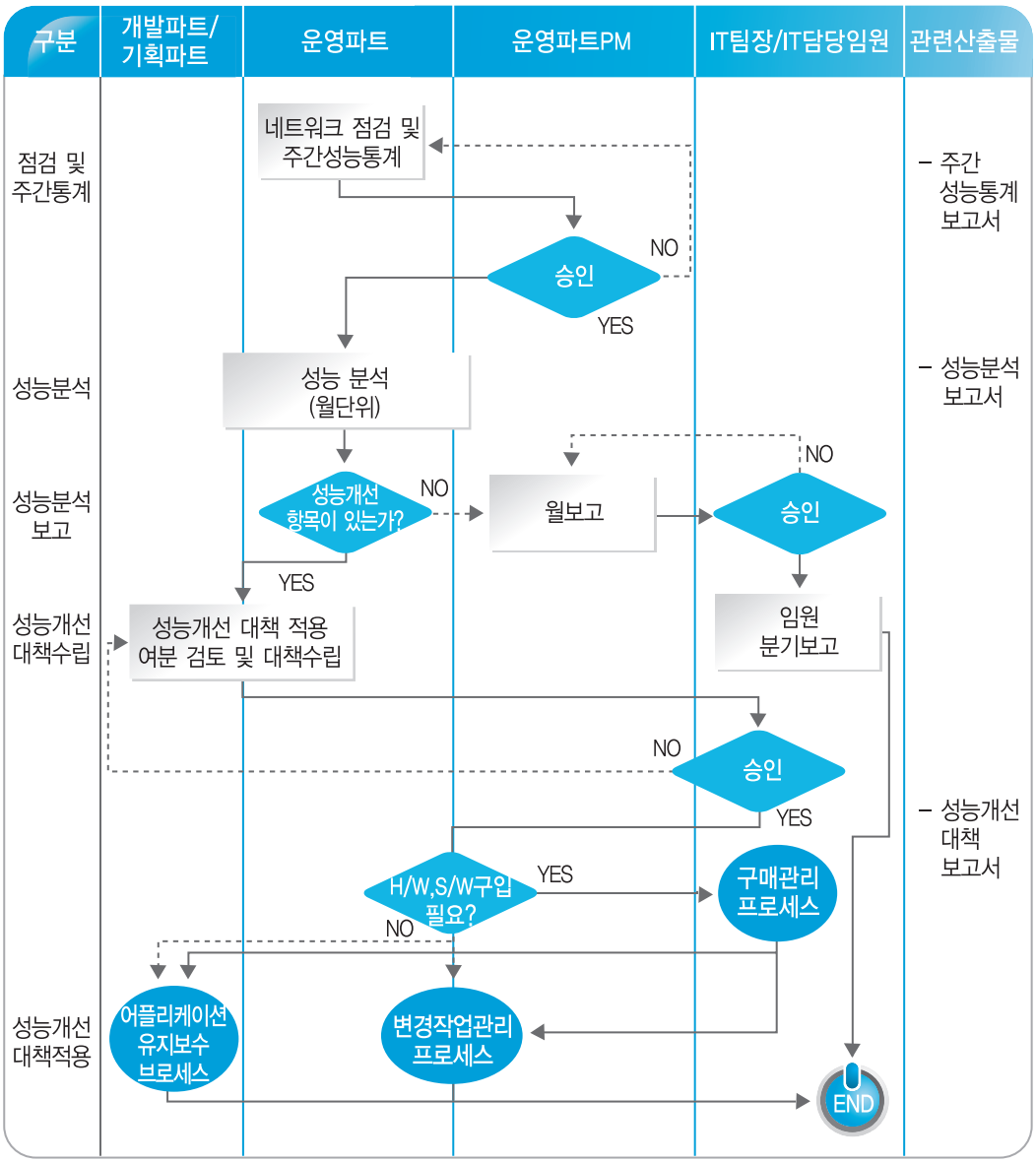
본 지침의 네트워크 성능관리 대상 및 범위는 〈표 3-4〉와 같으며 각 항목을 기준으로 네트워크 성능관리를 수행, 관리한다.

〈표 3-4〉 네트워크 성능관리 대상 및 범위

대상	내용
장비 성능관리	– 관리대상 장비의 CPU와 메모리에 대한 사용률(평균값, 최대값) 측정, 분석 – 관리대상 장비성능을 장비별, 지역별, 기간별(시간별, 일별, 주간별, 월별)로 측정, 분석 – 장비성능에 대한 임계치 관리 방식에 따른 경보체계 구축 및 관리 – IPS, IDS, F/W 장비의 S/W적인 문제(성능)는 배제하며, 장비자체의 성능 측정에 한한다.
세션 성능관리	– 조회 조건에 따라 각 스위치 장비의 세션 추이를 측정, 분석 – 스위치별 세션(SLB, FLB, Redirection) 생성시간 및 세션 수량을 측정, 분석 – 스위치별 세션(SLB, FLB, Redirection) 현황을 장비별, 지역별, 기간별로 측정, 분석
회선 성능관리	– 네트워크 관리의 핵심사항이 되는 장비별 회선에 대한 성능을 사용율을 기준으로 측정, 분석 – 장비별 회선 Bandwidth 측정, 분석 – LAN, WAN상의 Packet Error율, Discard율 측정, 분석 – 회선 성능에 대한 임계치 관리 방식에 따른 경보 체계 구축 및 관리
응답시간 관리	– 관리대상 장비의 평균 응답시간, 최대 응답시간, 평균 Packet Loss율, 최대 패킷 손실율(Packet Loss Rate) 측정, 분석 – 관리대상의 지역별, 장비별, 기간별(시간별, 일별, 주간별, 월별)로 분석

다. 성능관리 흐름도

네트워크 점검 및 성능관리를 위하여 각 담당별, 단계별 흐름도는 (그림 3-9)와 같다.



(그림 3-9) 네트워크 점검 및 성능관리 흐름도

3 2.3 DBMS

가. 개요

대부분의 조직은 기간업무 및 대 고객 서비스를 위해 데이터베이스를 사용한다. 이러한 업무들을 지원하는 중요한 데이터베이스 시스템은 시간이 지남에 따라 데이터 및 사용자의 증가로 인하여 성능이 점차 악화되어 가는 것이 일반적이다.

데이터베이스 성능관리는 이러한 시스템의 효율 및 응답속도 등을 최적의 상태로 유지 및 제공하기 위하여, 낮은 성능 현상을 보이는 요소를 찾아 성능 개선을 수행하거나, 성능 분석을 통하여 문제점을 발견하여 개선하는 것을 말한다. 이 같은 성능관리 업무는 다각적인 분석 및 모니터링, 그리고 이에 대한 적절한 조치가 병행되어야 하며, 이를 위한 성능관리 절차(프로세스)가 정립되어 있어야 한다.

대부분의 비즈니스 서비스 환경이 기존의 기본적인 고유 업무에 머무는 것이 아니라 다양한 형태의 정보 및 데이터 분석을 통해 그 결과를 업무에 적극 활용함으로써 최상의 생산성을 추구하는 방향으로 급격히 변화하고 있다. 이들의 정보 욕구는 갈수록 다양화, 양질화되어 가는 추세에 있으므로 관리해야 할 데이터의 대용량화 역시 필연적일 수밖에 없다. 이미 데이터 관리 용량은 GB(Giga Byte)에서 TB(Tera Byte)로의 VLDB(Very Large Database) 시대에 들어서 있는 것이 현실이다.

또한 전산시스템의 환경도 2Tier와 3Tier(Middleware 사용)가 공존하며 경우에 따라 업무적으로도 OLTP와 Batch가 병행되는 대단히 복잡하고 난이한 운영이 요구되고 있으므로 이를 밑받침하는 다양한 기술의 제공과 활용이 적절하게 수반되어야 한다. 더구나 하드웨어 및 소프트웨어의 구성 환경도 단순한 단일 노드의 구조에서 H/W 벤더들이 제공하는 낮은 수준의 HA(High Availability) 구성 또는 데이터베이스 업체에서 제공하는 병렬서버(Parallel Server)로의 구성을 통해 시스템의 고가용성을 보장하여 운영 업무의 연속성(24*7)을 강화하려는 방향으로 이미 전환된 상황이라 할 수 있다.

이러한 대용량 시스템을 구축하고 운영하면서 가장 중요한 부분으로 직면하게 되는 것은 성능관리와 안정성의 문제이다. 왜냐하면 대용량 데이터베이스 시스템의 구축시 데이터모델링, 물

리적 DB 구조, 데이터베이스 구성, 응용 프로그램, 하드웨어(서버), 네트워크 등 각 부문별 성능관리 관점의 고려요소가 철저하게 반영되지 않고서는 만족스러운 성능을 보장 받을 수 없으며, 이러한 성능저하의 요소는 곧 시스템의 안정성 확보에 좋지 않은 영향을 미치게 되어 미션 크리티컬(Mission Critical)한 정보 서비스의 연속성에 결정적인 저해요소로 작용하기 때문이다.

따라서, 대용량 데이터베이스 시스템을 구축하는 경우 성능관리 차원에서 각 부문별 부하요소의 최소화를 위한 전략적인 설계가 필요하며, 실제 구현단계와 운영단계에서 효율적인 성능관리 프로세스의 적용 여부가 서비스의 성공을 좌우하게 된다.

(1) DBMS 성능관리

성능관리란 데이터베이스 서비스를 제공하기 위해 구성된 각 구성요소에 대한 성능관련 자료를 수집 또는 실시간 감시를 통해 분석하고, 문제영역을 식별, 그에 대한 원인을 분석하며, 이를 근거로 튜닝을 실시하여 최적의 데이터베이스 상태를 유지하도록 하는 일련의 행위를 말한다.

- 설계단계부터 적용 : DBMS 성능을 최적의 상태로 유지하기 위하여 응용프로그램 또는 프로젝트 전 과정에 걸쳐 성능관리가 이루어져야 한다. 따라서 설계 단계에서부터 이를 고려하여야 한다.
- 목표 설정 : 투자대비이익률(ROI : Return on Investment)을 고려한 방법론을 적용하여 성능관리 집중 대상 프로그램이나, 프로젝트를 선별하는 것이 권장된다. DBMS 성능관리에서는 일반적으로 80/20 법칙을 적용하여 성능관리 및 성능조정을 고려한다. 트랜잭션의 20%가 전체 시스템 자원의 80%를 사용하기 때문이다.
- 성능 목표 이행상황 모니터 : 목표를 정하고 동의하였으면 이 목표를 달성하기 위한 성능관리의 시작으로 모니터링을 실시한다. 성능 관련 데이터의 수집은 단기적인 현재 상황의 분석에서 장기적인 성능 및 용량 예측을 위하여 자세하게 기록해야 하며, 정기적으로 측정값을 공표해야 한다.
- 공동 작업 : 데이터베이스 관리자, 응용 프로그램 설계자/개발자, 시스템(네트워크) 관리자가 공동으로 성능관리 작업에 참여하여야 한다.

(2) DBMS 성능관리의 목표

성능관리의 목적은 데이터베이스의 자원, 성능현황 데이터를 수집, 관리함으로써 경향을 분석하고, 시기적절한 용량계획을 수립할 수 있도록 하는 것이다. '예상 장애에 대한 선 조치 항목 식별', '용량계획의 기초 데이터를 제공함으로써 효율적인 투자계획 수립가능', '각 자원에 대한 추이분석' 등의 사전 목표를 확립한 후 성능관리 대상에 대한 모니터링과 분석, 조정이 진행되어야 한다.

또한, 데이터베이스 성능관리의 구체적인 목표는 응용 프로그램 사용자가 해당 명령문에 대한 응답을 최대한 빨리 얻을 수 있도록 하는데 있다. 응답시간은 정량화 가능한 것이어야 하며, 이 세부 지표(목표)는 다음의 관점에서 측정되어야 한다.

- 대기 감소 및 제거
- 가장 적은 수의 디스크 블록 액세스
- 메모리에 블록 캐시
- 응답시간(Response Time)
- 처리능력(Throughput)
- 워크로드(Workload)

(3) DBMS 성능관리의 원칙

- 미리 정의된 성능기준치 이상의 효율을 갖도록 한다.
- 튜닝을 위한 자료의 취합 및 튜닝방법, 결과분석에 따른 조치방법 등, 기본적인 관리절차에 준하여 관리한다.
- 분석자료의 취합 방법은 성능에 가장 영향을 미치지 않는 방법을 선택한다.
- 자료분석에 의해 조치가 완료되면, 성능향상 유무를 확인하고 변동사항에 대하여 이력관리를 수행한다.
- 일일 단위, 주 단위, 월 단위로 구분하여 주기적이고 반복적인 성능관리를 실시한다.

(4) 성능관리의 영향요소

데이터베이스의 성능 영향요소는 시스템 자원인 CPU, 메모리, 디스크, 네트워크 부분과 직접적인 상관관계가 있으므로, 성능분석 시 다음 관련 자료를 상호 검토하여야 한다.

- 시스템 관리자, 어플리케이션 담당자, 데이터베이스관리자(DBA)가 필요하다고 판단하는 기타 항목에 대해 별도의 기준치(Base Line) 설정 후 관리한다.
- 일시적인 성능 조정(튜닝)의 목적으로 특정 항목에 대한 관리가 필요할 경우라 하더라도 관리 항목의 성능관리 절차에 준하여 관리한다.

(5) DBMS 성능관리의 담당자

DBMS의 성능관리는 데이터베이스 관리자만의 영역이 아닌 프로젝트 상의 DBMS 이용 및 관리자들이 모두 해당된다. 응용 프로그램 설계자의 데이터베이스 사용에 관련된 로직과 모델링(Modeling), 그리고 응용 프로그램 개발자의 SQL 문장은 전체 데이터베이스 성능의 70~80%를 좌우한다. 시스템 관리자의 자원(CPU, 메모리, 디스크 I/O, 네트워크) 관리 영역의 영향도 역시 DBMS 성능관리에 적극적으로 반영되어야 한다.

〈표 3-5〉 DBMS 성능관리 수행자와 역할

DBMS 성능관리 담당자	성능관리 역할
업무분석가	- 업무 프로세스 최적화
응용 프로그램 설계자	- 데이터 설계(정규화, 반정규화) - 프로세스 설계(어플리케이션 로직)
응용 프로그램 개발자	- SQL 문장 튜닝 - 물리적 구조 튜닝
데이터베이스 관리자	- 메모리 할당 튜닝 - I/O 경합 튜닝 - 메모리 경합 튜닝
시스템 관리자	- 운영체제 및 H/W 튜닝

(6) DBA의 주요 성능관리 업무

일반적으로 DBMS 성능관리의 담당자는 데이터베이스 관리자가 수행하게 된다. DBA는 DBMS 성능관리를 위하여 성능관리 수행자들의 의견을 조율하며, 다음 내용들을 계획에 반영하여야 한다.

- ▶ DBMS 성능관리 대상 요소 및 측정 항목
- ▶ 측정 항목별 임계치 설정
- ▶ 임계치를 초과하는 항목에 대한 분석 방안
- ▶ DBMS 성능 측정 항목에 대한 측정 방법 및 측정 주기
- ▶ DBMS 성능 측정 결과를 이용한 장기적 성능 분석 방안

또한, 위 내용에 대한 구체적인 측정 대상별 성능 추이 분석 결과 및 통합된 관점의 분석 결과가 명시되어야 하며, 다음 사항이 포함되어야 한다.

- ▶ DBMS 성능 분석 대상의 구성 정보(업무용도, 대상명, 설치 위치, 구성 항목 등)
- ▶ DBMS 성능 측정 도구 및 측정 방법 정보
- ▶ DBMS 성능 측정 결과에 대한 성능 및 추이 분석
- ▶ 통합된 관점의 성능 분석 결과 : 전체 IT 서비스에 미치는 영향 판단

나. DBMS 성능관리 대상 및 범위

본 지침에서 제시하는 DBMS의 성능관리의 대상 및 범위는 각 기관에서 운영하는 DBMS의 안정적이며, 영속적인 서비스의 제공을 위하여 크게 다음의 영역으로 분류하여 적용한다. 이 영역들은 상호 의존적이며 서로 유사한 기술군의 영역으로 결합된다.

이러한 성능관리의 대상들은 성능 지표의 설정 및 지속적인 모니터링과 에러 체크, 분석을 통하여 항상 최상의 성능상태를 유지할 수 있도록 관리되어야 한다. 본 지침에서는 DBMS 성능관리

대상 및 범위를 다음 〈표 3-6〉과 같이 분류하며, 각 항목을 기준으로 DBMS 서비스의 구체적인 환경(특성)에 맞게 성능관리 지침서 및 절차서에 의한 관리 계획을 수립한다.

〈표 3-6〉 DBMS의 성능관리 대상 및 범위

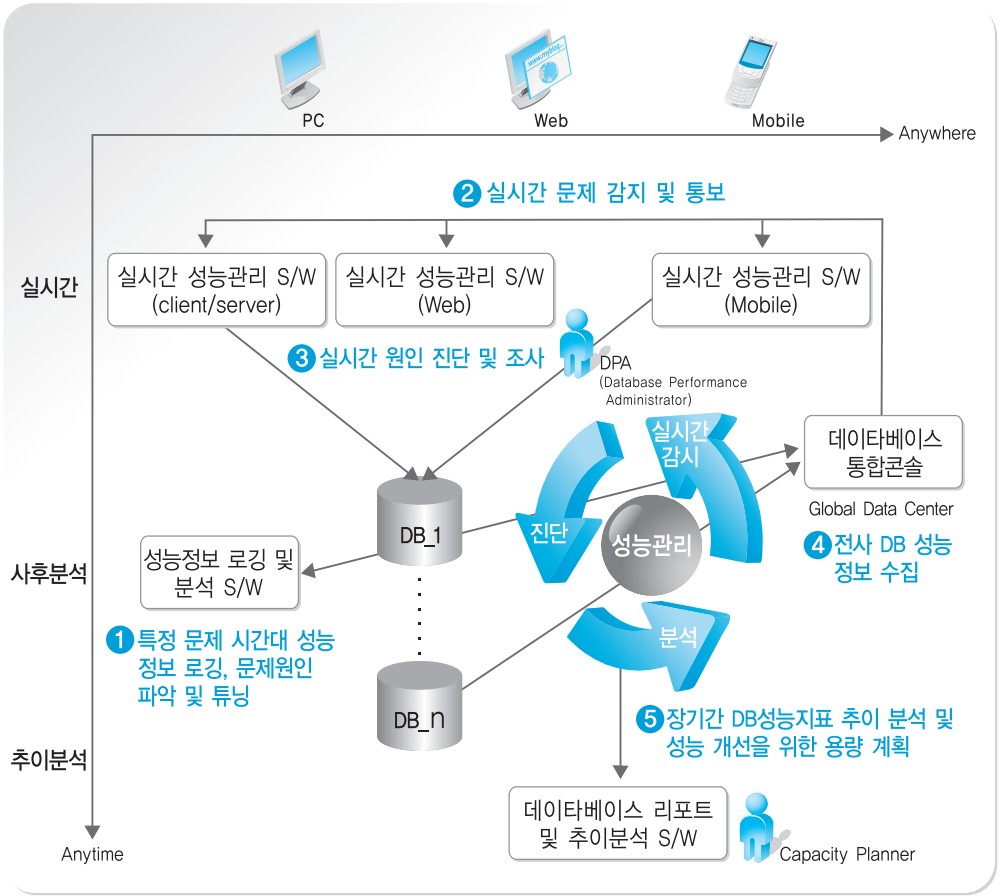
대상	범위
스키마(schema)	스키마에 대한 성능관리는 데이터의 물리적인 구조를 통하여 이루어진다. – 테이블(모델링)과 인덱스의 설계 및 생성, 삭제 – 테이블의 정규화(Normalization)를 통한 중복 배제 – 테이블의 반정규화 (De-Normalization)를 통한 성능극대화
응용 프로그램 (문장튜닝)	응용 프로그램 영역에 대한 성능관리는 업무 기능 및 이 기능을 구현하는 프로그램 모듈을 다룬다. – 응용 프로그램 유형에 맞는 프로시져 코드의 분석, 운영, 튜닝 – 어플리케이션으로부터 수행되는 SQL 문장 분석, 튜닝 – 어플리케이션의 로직 조정을 통한 성능 개선점 분석
공유메모리 영역 (파라미터튜닝)	공유메모리 영역은 적절한 크기의 캐쉬 풀(Cache Pool)을 운영관리하여 성능의 극대화를 유도한다. – 데이터 버퍼 영역을 설정하여 OLTP 업무의 경우 90% 후반대의 데이터 캐쉬 적중률을, 데이터웨어하우스의 경우 60% 이상의 캐쉬 적중률을 유지 – SQL 문장의 처리를 위한 공유메모리 영역의 운영을 통하여 하드파싱을 방지하고 소프트파싱을 유도 – 기타 메모리 영역의 운영을 통하여 추가적인 CPU의 사용 및 Disk I/O량을 조정한다.
데이터베이스 (물리적인 데이터 파일 관리)	디스크에서 데이터베이스 파일의 물리적인 분산 배치를 통하여 I/O 성능을 관리하는 영역으로 데이터베이스 병목현상의 가장 많은 부분을 차지하는 I/O 성능을 관리한다. – 데이터 파일의 분산배치 – 컨트롤 파일의 분산배치 및 이중화 – 리두로그 파일의 분산배치 및 이중화 – 아카이브 모드의 데이터베이스 운용 – 세그먼트(테이블, 인덱스 등)의 단편화(Fragmentation) 관리

Locking 관리	데이터베이스 기본 잠금(Locking) 방식은 최고의 데이터 동시성을 허용하면서 데이터의 일관성을 보장하기 위한 수단으로써, 성능관리의 주요 범주에 포함되어야 한다. - 잠금 유형별 어플리케이션의 사용(Row, Block, Table 레벨의 잠금 판단 및 운용) - 잠금 경합의 원인 및 진단 - 잠금 경합 및 교착 상태의 해결
사용자 기대 수준	성능관리자는 사용자들이 요구하는 기대 수준을 분석/기록하여 항상 이에 대응하는 성능 상태를 유지하여야 한다. 사용자들의 기대 수준은 설계 및 구축 단계 이전에 분석되어 DBMS 시스템에 반영되지만 일반적으로 운영과정을 거치면서 변하게 된다. 따라서 일관성 있는 사용자 기대 수준을 충족시키기 위하여 SLA를 통한 정량화된 서비스 만족도를 파악하여야 한다. 결국 사용자의 기대 수준 또한 DBMS 성능관리의 대상으로 볼 수 있음을 의미한다.

다. DBMS 성능관리 구성도

데이터베이스 관리자는 PC, 웹, 휴대폰/ PDA 단말기를 통해서 언제든지 자신이 관리하는 데이터베이스에 실시간으로 성능을 관리할 수 있어야 한다. 또한, 데이터베이스 관리자는 실시간 성능 모니터링뿐만 아니라 특정한 성능 문제를 반복적으로 야기하는 시간대의 시스템 전체 성능 정보 또는 특정 프로그램의 성능 정보를 저장하여 심도 있는 사후 분석, 시스템 전체의 성능 추이 분석을 통한 DBMS 용량 계획 등 필요한 성능관리 업무를 수행할 수 있어야 한다.

전반적인 DBMS 성능관리의 논리적 수행 단계는 (그림 3-10)의 순서대로 진행된다. 우선 데이터센터에서는 전사 e-business의 구성 데이터베이스의 주요 성능 지표(DBMS의 건강상태 지표)를 실시간으로 모니터링 한다. 모니터링 과정에서 특정 데이터베이스의 성능에 문제가 발생하면, 응급조치와 더불어 담당 데이터베이스 관리자에게 즉각적인 연락을 취하게 된다.



(그림 3-10) DBMS 성능관리를 위한 아키텍처

자신이 관리하는 업무와 DBMS 시스템의 특징에 익숙한 데이터베이스 관리자는 대부분의 문제를 실시간으로 해결할 수 있지만, 새로운 업무 패턴이나 데이터량의 변화에 따라 새로 출현하는 성능 저하 현상에 대해서는 실시간으로 문제를 해결할 수 없는 경우도 있다. 이와 같은 상황을 위해서는 데이터베이스 성능 정보를 저장하고 이를 사후에도 철저하게 해부할 수 있어야 한다.

잘 설계되고 최적으로 성능관리된 DBMS라 하더라도 급격히 늘어나는 데이터와 사용자 수의 변화로 인하여 성능문제가 촉발할 수도 있다. 이와 같은 상황에서 최적의 비용으로 DBMS의 성능

을 유지하기 위해서는 데이터베이스 관리자가 장기간에 걸쳐 DBMS 성능 지표의 추이를 분석하고 용량을 계획할 수 있도록 지원하는 전략이 필요하다. 따라서 DBMS 성능관리 책임자와 성능관리 담당자는 잘 계획된 프로세스와 철저한 분석을 통하여 항시 최적의 성능을 제공할 수 있는 시스템 상태를 유지하여야 한다.

라. 도구를 이용한 DBMS 성능관리

DBMS 관리 도구를 이용하여 성능관리 프로세스를 자동화한 경우 다음과 같은 필요(충분)기능이 제공되어야 DBMS 성능관리 아키텍처를 올바로 지원할 수 있다.

(1) 실시간 모니터링 지원

- 연동되는 여러 데이터베이스들을 한 화면에 모니터링하면서 문제가 발생하는 데이터베이스를 빠르게 판별할 수 있어야 한다.
- 데이터베이스 상에서 발생하는 성능문제를 실시간에 감지하여 문제의 원인이 되는 세션과 SQL을 빠르게 찾아내어야 한다.
- 특정 배치작업 또는 프로그램 등이 세션으로 접속되어 수행되는 현황을 실시간에 모니터링 해야 한다.
- 락(Lock)의 대기(Wait) 현상 발생시 즉시 Holder 세션과 Waiter 세션의 접속정보, 유발 SQL, 대상 오브젝트, Holder의 현재 수행정보, Waiter의 대기시간 등을 발견할 수 있어야 한다.
- 자원의 사용현황과 자원의 경합현상, 유발시킨 SQL들을 유기적으로 모니터링 해야 한다.
- O/S의 자원사용정보와 데이터베이스의 성능정보를 유기적으로 결합하여 모니터링 해야 한다.

(2) 성능정보의 채집 및 기록

- 성능정보의 채집 및 기록하는 과정에서 데이터베이스가 운영되는 서버에 미치는 부하가 최소화되어야 한다.
- 성능정보의 채집 활동을 스케줄링하고, 채집주기를 설정할 수 있어야 한다.

- 채집된 성능정보를 기록하여 사후(데이터베이스 재기동, 세션 종료) 이후에도 실시간의 수행상황과 거의 유사하게 재생하면서 성능문제의 원인을 밝힐 수 있는 데이터를 제공해야 한다.

(3) 경보 기능

- 자원의 과다 점유 발견 시에 즉시 경보해야 한다.
- 경보 이후에 사후 조치의 근거가 되는 정보들을 자동으로 기록하여야 한다.
- 테이블스페이스 또는 롤백세그먼트에 대한 사용 현황을 경보하여야 한다.

(4) 튜닝 지원

- 과부하를 유발하는 SQL들을 판별하여야 하고, 그러한 SQL들에 대해서 과부하의 원인을 찾기 위해 수행계획(플랜)을 확인하고 액세스되는 테이블과 인덱스의 현황을 조회할 수 있어야 한다.
- 성능문제를 유발하는 SQL들에 대해서 튜닝된 대안 SQL의 플랜 및 수행비용(cost)들을 비교 분석할 수 있어야 한다.
- 인스턴스의 파라미터 현황을 조회하고, 현재 성능정보를 통해 잘못 구성된 파라미터에 대해 변경 가이드를 제시하여야 한다.
- 디스크별 I/O 현황을 통해 특정 데이터파일에 I/O가 치중 되는지에 대한 정보를 제공해야 한다.

(5) 기타

- 벤더사의 데이터베이스 관련 기술력(튜닝 포함)이 뛰어나야 한다.
- 대규모의 VLDB 사이트에서 적용되어 검증된 사례가 있어야 한다.

3 2.4 응용 소프트웨어

가. 개요

3.1.1절에서 언급되었듯이 최근의 정보시스템들은 조직의 핵심 업무 처리를 담당하고 있을 뿐만 아니라 경쟁 우위 확보를 가능하게 하는 전략적 도구로서 자리 잡아 가고 있다. 이러한 시스템들의 성능이 저하되거나 장애가 발생하게 되면 적절한 업무 처리가 불가능하게 되고 경우에 따라 상당한 금전적 손실을 가져오게 된다. 이러한 시스템들을 미션 크리티컬(mission-critical) 시스템이라고 하며 뱅킹시스템, 항공/철도 예약 시스템, 전자상거래 시스템, 의료 및 안전/재난관리 시스템 등이 해당된다.

미션 크리티컬 시스템들의 대표적인 특징은 조직 내부 사용자 외에 불특정 다수를 대상으로 하며 시스템을 구성하는 하드웨어, 소프트웨어, 어플리케이션, 네트워크 장비들이 단일 지역이 아닌 광범위한 지역에 배치되어 있고, 일일 24시간 무중단으로 운영되어야 한다는 것이다. 따라서, 개별적인 서버, 라우터, 데이터베이스 등의 운영 정보 보다는 전체 업무 어플리케이션의 관점에서 목표했던 서비스 수준이 유지되고 있는지를 파악하는 것이 보다 중요하다고 볼 수 있다. 즉, 현재 업무처리 트랜잭션이 성공적으로 처리되고 있는지, 응답 시간이 적정한지, 시스템을 사용하고 있는 사용자는 누구이고 어느 정도 규모인지 등을 관리할 수 있는지의 여부가 조직의 업무를 성공 또는 실패로 이끌 수 있는 열쇠가 되고 있다.

현재 가장 일반적인 업무 어플리케이션 모델은 N-티어(Tier) 클라이언트/ 서버 모델이라고 할 수 있는데, 이는 전체 어플리케이션의 기능을 그 특성에 따라 둘 또는 그 이상의 티어(tier)로 분리하여 구성하는 것으로 업무 처리 서비스 또는 정보를 제공하는 서버 부문은 하나의 시스템 또는 다중의 시스템들에 배치되고, 경우에 따라 외부 시스템들과도 실시간으로 정보를 교환하며, 서버에서 제공하는 서비스 또는 정보를 이용하는 클라이언트 부문은 다양한 미디어(PC, 전용 단말기, 모바일 등)에 탑재되어 운영된다.

클라이언트 어플리케이션이 요청 또는 응답 정보를 처리하는 동안 사용자는 필요한 서비스나 정보를 어떤 경로로 얻어오는지 알지 못한다. 클라이언트 어플리케이션은 네트워크 상에 분산되

어 있는 여러 서버들의 서비스를 이용하지만 사용자는 업무 처리에 대한 응답 시간이 평소보다 길어지거나 몇 번의 재시도에도 정상적인 처리가 되지 않는 경우 시스템의 무엇인가가 잘못되었을 것이라고 짐작만 할 뿐이다. 사용자가 시스템 담당자 또는 서비스데스크에 이러한 사실을 통보 또는 문의하는 경우에는 시스템의 문제점을 확인하여 개선할 수 있는 기회가 확보되지만 많은 사용자들은 불평만 하다가 그냥 지나쳐 버리거나 여러 번 반복될 경우 해당 시스템을 더 이상 사용하지 않게 된다.

따라서, 미션 크리티컬한 정보시스템을 효율적으로 운영하기 위해서는 사용자 관점에서 업무 어플리케이션의 응답시간이 지연되거나 트랜잭션 처리가 실패하는 경우 그 원인을 정확히 찾아내어 신속히 조치하는 것이 필수적이며, 이를 효과적으로 수행하는 것이 응용 소프트웨어 성능관리의 목표이다. 과거에는 정보시스템이 단순하여 하드웨어 또는 프로그램의 작동 여부를 감시하는 것만으로도 충분한 성능관리가 이루어졌으나, 현재와 같이 복잡한 시스템 구조 하에서는 작동 여부 감시 외에 서비스의 처리 과정에 직접적으로 관련되어 있는 쓰레드 및 컴포넌트에 대한 응답시간 측정, 시간당 처리량 측정, 각종 이벤트 및 대기 큐들의 상태감시도 중요한 부분이 되었다.

〈표 3-7〉은 성능 관련 문제의 해결 과정에 따른 전통적인 접근 방법과 최근의 접근 방법을 비교한 것이다.

〈표 3-7〉 성능 관련 문제 해결 과정에 따른 접근 방법 비교

문제해결단계	전통적 접근 방법	최근의 접근 방법
운영 환경	– 담당자별 관리도구, 측정 지표 상이	– 공통의 서비스 측정지표 사용
문제 인식/확인	– 수동적(Reactive) – 사용자의 연락을 통해 문제 확인 및 우선 순위 결정 → 많은 경고 메시지들이 인식되지 않고 지나침 – 시스템 차원의 경고 메시지 – 영향을 받는 어플리케이션 유추 필요 →성능 문제의 72%는 모니터링 도구에서 발견되지 않음(Forester)	– 능동적(Proactive) – 주요 서비스에 대한 적극적 감시 → 문제 예방 가능, 비즈니스 목표에 따른 우선 순위 결정 – 비즈니스 차원의 경고 메시지 – 비즈니스 영향도/심각도 확인
문제 영역 확인	– 모든 담당자 확인(Brute Force) – 모든 담당자들이 해당 분야의 문제 발생 여부 확인 → 자원(시간, 비용) 낭비로 인해 비즈니스 영향 시간 증가 – 책임 회피(Not me syndrome) – 개별 시스템들은 작동되고 있음	– 해당 담당자 확인 – 문제 선별, 문제 영역 확인 시간 단축 및 올바른 담당자 투입 가능
문제 분석	– 시험/경험/운에 의존 – 개인적 지식 의존 및 과거 사례를 모두 동원 → 시간 낭비로 인해 비즈니스 영향 시간 증가	– 부하 모델/트랜잭션 기반 – 서비스에 영향을 주는 트랜잭션 식별 및 해당 트랜잭션의 자원 사용량 분석
문제 해결	– 수동 조치 – 수동 조치 및 라인 단위 코드 분석 → 반복적으로 발생되는 문제에 대하여도 시간 낭비	– 자동 조치 – 반복되는 문제의 자동 조치, 복잡한 해결 작업들의 최적화 가능

나. 성능관리 대상 및 범위

응용 소프트웨어는 크게 응용 프로그램, 응용 플랫폼, 응용 솔루션으로 구분할 수 있으며 각각의 정의, 성능관리 대상 및 범위는 다음과 같다(자세한 설명은 4.2.4절, 응용 소프트웨어를 참조한다).

(1) 응용 프로그램

응용 프로그램은 정보시스템 구축 또는 운영 시 프로그래밍 언어를 사용하여 신규 또는 변경 개발된 모듈을 의미하며 응답 시간, 시간당 트랜잭션 처리량이 주요 관리 대상이다. 응용 프로그램의 구간별 응답 시간은 일반적으로 프로그램 내에 관련 코드를 작성하여 로그 파일에 저장되도록 하지만 운영 시에는 성능 향상을 위해 디버깅 레벨의 로그를 남기지 않도록 하는데, 이럴 경우에도 성능상의 문제점이 발견되면 쉽게 확인이 가능하도록 로그 레벨을 동적으로 변경하여 적용할 수 있도록 하여야 한다. 상용 APM(Application Performance Management) 도구 등을 적용하는 경우에는 자체 지원 기능 (Introscope, JVMPI 등)에 의해 구간별 응답 시간을 확인할 수 있다.

〈표 3-8〉 응용 프로그램의 성능관리 대상 및 범위

대상	범위
응답 시간/ 배치 실행 환경	– 부하량에 따른 응용 프로그램의 평균, 최소, 최대, 90% 응답 시간 측정 및 분포도와 대기 시간 분석 및 튜닝 – 부하량에 따른 응용 프로그램의 초당 트랜잭션 처리 건수, 전체 처리 건수 측정 및 Throughput 변화 분석 – 함수 또는 메쏘드 호출의 타임아웃 적용 여부, 타임아웃 발생 건수 측정, 타임아웃 발생시의 응답 시간 확인, 원인 분석 및 튜닝
메모리 사용	– 응용 프로그램 코드 및 라이브러리에 대한 메모리 크기 측정 – 시간당 또는 일별 메모리 증가량 및 증가율 측정, 원인 분석 및 튜닝
데이터베이스 처리	– 커넥션 풀(Connection Pool) 사용 여부, Cursor 유형 및 bind 변수 사용 여부, 사용된 SQL 및 SQL 조회 처리의 효율성 분석
오류 및 예외	– 응용 프로그램의 오류 및 예외 발생 건수, 유형 및 패턴 분석
배치 실행 환경	– 배치 프로그램의 수행 계획, 수행 시간, 선·후행 작업 분석

(2) 응용 플랫폼

응용 플랫폼에는 TP Monitor, WAS(Web Application Server), 웹 서버, EAI(Message Queue 등) 등 응용 프로그램 실행을 위한 기반 환경 및 서비스(트랜잭션, 보안, 네이밍, Persistence 등)를 제공하는 미들웨어 제품이 포함된다. 대부분의 경우 〈표 3-9〉와 같은 기준을 적용할 수 있으나 동일 계열(예: WAS)의 제품이라고 하더라도 실제 도입되는 제품별로 세부적인 관리 대상 및 측정 기준에는 차이가 있으므로 각 기관에 도입된 솔루션 및 시스템 환경에 따라 적합한 기준과 관리 절차를 마련해야 한다.

〈표 3-9〉 응용 플랫폼의 성능관리 대상 및 범위

대상	범위
응답 시간/ 트랜잭션 처리량	– 부하량에 따른 평균, 최소, 최대 및 90% 서비스 시간 측정, 분포도와 대기 시간 분석 및 튜닝 – 부하량에 따른 초당 트랜잭션 처리(요청) 건수, 전체 요청(처리) 건수 측정 및 Throughput 변화 분석 – 발생 건수, 타임아웃 발생 시의 응답 시간 측정 및 원인 분석, 튜닝
대기 큐/ 대기 시간	– 대기 큐에 저장된 요청들의 실시간, 평균, 최대 요청 개수 측정 및 원인 분석 – 대기 큐에서 소요된 평균, 최대 대기 시간 측정, 원인 분석 및 튜닝 – 대기 큐에 저장되지 못하고 거절된(rejected) 요청들의 개수 및 발생
프로세스(쓰레드) 상태 및 개수	– 프로세스 또는 쓰레드의 실시간 서비스 상태 및 자원 (CPU, 메모리, 디스크 I/O 등) 사용량, 메모리 사용량 추이 및 분석 – 시간당 GC(Garbage Collection) 횟수 및 평균, 최대 처리 시간, GC 전, 후의 메모리 사용량 변화 측정, 분석 및 튜닝(J2EE, .NET) – 어플리케이션 프로세스 또는 쓰레드의 평균, 최대 실행 개수 측정, 및 분석
세션 상태 및 개수	– 클라이언트 연결 세션의 메모리 사용량 측정 – 클라이언트 연결 세션의 평균, 최대 개수 측정
통신 큐, 채널 상태	– 클라이언트 연결 요청에 대한 시간당 drop 건수 측정, 백로그(backlog) 튜닝 – TCP, UDP 등의 시간당 오버플로우(over-flow) 건수 및 비정상 TCP 연결(SYN_SENT, FIN_WAIT_2, CLOSE_WAIT) 건수 측정, 원인 분석 및 튜닝 – 사용하고 있는 파일 디스크립터(descriptor) 개수 측정

자원(Resource) Pool	– 데이터베이스 커넥션 풀(Connection Pool), 쓰레드 또는 객체 풀 등의 평균, 최대 사용 개수 측정, 분석 및 튜닝
오류 및 예외	– 응용 플랫폼의 오류 및 예외 발생 건수, 유형 및 패턴 분석
부하 분산	– 클라이언트 요청에 대한 실시간 부하 분산 상태 측정

(3) 응용 솔루션

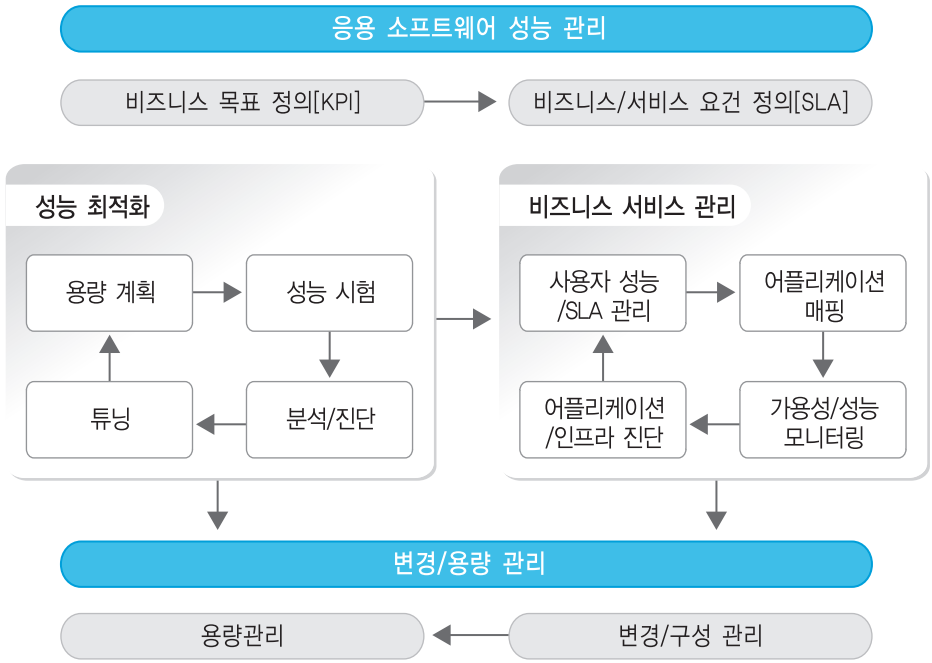
응용 솔루션에는 그룹웨어, CRM, ERP, SCM 등 업무 처리를 지원하는 비즈니스 솔루션과, B2Bi, BPM, ETL, EII 등 조직 내외부의 프로세스 또는 정보를 통합하기 위한 통합 솔루션 및 디렉토리, 인증 등의 보안 솔루션 등 다양한 도구들이 포함된다. ERP의 경우 최근 공공 기관에 활발하게 도입되고 있는데, 대부분의 솔루션은 자체적인 성능관리 기준 및 관리 모듈을 제공하고 있으므로 각 기관에 도입된 솔루션 및 시스템 환경에 따라 자체적인 기준과 관리 절차를 마련해야 한다. 〈표 3-10〉은 어느 솔루션의 경우 적용할 수 있는 성능관리 대상 및 범위의 예시이다.

〈표 3-10〉 응용 솔루션의 성능관리 대상 및 범위

대상	범위
구간별 수행 시간	– 평균 응답 시간, 대기 시간, Roll 시간, 로드 시간, 데이터베이스 시간의 측정, 분석 및 튜닝
대기 큐	– 대기 큐에 존재하는 평균 요청 개수, 대기 시간 측정, 원인 분석 및 튜닝
메모리/버퍼	– 메모리 영역/버퍼(R/3, roll area 등)의 부족 또는 과다 여부 확인, 오브젝트 버퍼링 여부 검토 및 튜닝
오류 및 예외	– 응용 솔루션의 오류 및 예외 발생 건수, 유형 및 패턴 분석

다. 성능관리 구조도

(그림 3-11)은 응용 소프트웨어 성능관리 프로세스의 생명주기(Life Cycle)를 나타내는 것으로 비즈니스 목표를 명확하게 정의하는 것으로부터 출발한다.



(그림 3-11) 성능관리 구조도

설정된 비즈니스 목표에 따라 서비스 요건이 정의되고 이에 따라 어플리케이션의 성능과 가용성의 구체적인 지표를 설정하여 관련 항목들을 체계적으로 모니터링하고 측정 결과에 따라 성능 조정(튜닝) 또는 용량 증설 등을 수행하게 된다. 성능 조정(튜닝) 및 용량 증설 작업은 변경/구성관리 및 용량 관리 프로세스에 의해 승인되고 관리되며 비즈니스를 수행하는 동안 지속적으로 반복 수행된다.

3 2.5 통합적 성능관리

가. 개요

통합관리시스템(Enterprise Management System)이란 IT 시스템을 위한 성능관리 활동을 자동화 및 통합 콘솔을 통하여 통합적으로 관리하는 것을 말한다. 이는 IT 시스템을 위한 각종 관리 활동(모니터링, 장애관리, 변경관리, 성능/용량관리, 구성관리, 자산관리, 백업관리, 스토리지 관리, 보안관리, Job 스케쥴링, 운영이관, 서비스수준관리, 서비스데스크, 관리자를 위한 리포팅 등)을 지원하고, 관리 활동 상호간의 연동 및 통합관리 기능을 지원한다. 단, 완전히 새로운 통합관리시스템을 구축하기 보다는 가급적 기존에 사용하고 있는 도구들을 활용하면서 이들간의 연동을 강화하고, 전체적 관점에서 통합하여 관리기능을 강화하는 방향으로 추진한다.

분산 컴퓨팅 환경의 통합관리 문제는 다음과 같은 결론으로 귀착될 수 있으며, 본 지침에서는 다음 사항을 기반으로 하여 통합적 성능관리의 지침을 제시한다.

- 통합적(중앙 집중적) 관리 : 분산 컴퓨팅 환경은 전체적인 관점에서의 통합적 관리로 관리 수준을 향상시킨다.

- 정책(관리 규칙) 기반의 관리 : 관리자는 미리 정의한 이벤트, 장애, 성능, 네트워크 정책들을 사용하여 일관된 관리를 한다.

- 서비스 중심의 관리 제공 : 시스템 관리의 가장 기본적인 목표는 업무서비스를 중단 없이 지원하는데 있다. 즉, 자원 중심의 관리가 아닌 업무(서비스) 중심의 관리를 하며, 업무서비스와 컴퓨팅 자원간의 상호연관 관계를 정의한다. 따라서 컴퓨팅 자원의 장애가 업무서비스에 미치는 영향을 한 눈에 파악하게 되며, 신속한 장애복구가 가능하게 한다. 또한 업무 프로세스적인 관점에서의 관리를 통하여 업무 서비스의 상태를 한 눈에 파악할 수 있으며, 자원 중심의 관리뿐만 아니라 장애의 신속한 탐지, 장애의 영향 분석, 서비스수준관리를 보장한다.

나. 성능관리 대상 및 범위

(1) 적용 범위

정보시스템을 최적의 상태로 관리, 운영하기 위해서는 통합관리 체제뿐 아니라 운영을 위한 절차와 지침 그리고 우수한 인력이 필수적이며 통합 성능관리시스템은 그 범위가 넓어 단계적으로 접근하는 것이 적합하다. 통합적 성능관리 시스템은 반드시 모든 시스템에 공통된 수준으로 적용해야 되는 것이 아니라, 시스템별 관리 필요수준에 따라 그 적용수준을 차등화할 수 있다. 따라서, 시스템별로 운영관리의 수준을 정의할 때 가이드라인으로 활용할 수 있다.

따라서 모든 운영기능은 그 기능의 업무 특성과 시스템의 특성에 따라 혹은 운영기능 자체의 특성에 따라 적용되는 수준과 대상이 차별화되며, 통합성능관리의 적용 범위는 〈표 3-11〉과 같다.

〈표 3-11〉 단계별 통합적 성능관리 적용 범위

성능관리 기능 / 어플리케이션/서버 그룹		성능관리 적용범위 (관리수준 요구사항)
Group 1	기업의 핵심업무를 지원하며 일반고객 및 트랜잭션 처리 시스템 예) 영업, 수/미납, 청구	- 월 1회 성능 추이 분석, 성능 개선방안 수립/보고
Group 2	기본적인 경영관리 및 업무지원을 위한 시스템 예) 재무, 회계, 그룹웨어	- 서버, DB, 어플리케이션, 네트워크 성능에 대한 임계치 정의 및 모니터링 - 월 1회(또는 SLA에 규정된 시기에) 성능 추이 분석 수행 - 성능 이슈 발생시 성능개선을 위한 튜닝, 구성변경 작업 수행 - 필요시 통합 성능 테스트 및 결과보고
Group 3	경영정보 지원 및 의사결정 지원 시스템 예) 인사, Data Mining, EIS	- 성능 이슈 발생시 성능 개선작업 수행
Group 4	모든 시스템의 개발/시험용도로 사용 중인 서버 예) 개발용 시스템/서버	- N/A - 상위그룹은 하위그룹의 서비스를 포함함

(2) 적용 대상

(가) 적용 대상 선정기준

통합성능관리시스템으로 관리해야 할 어플리케이션/서버를 선정할 때에는 어플리케이션의 업무 특성과 서버의 IT 특성을 고려하여 대상을 선정하여 각 시스템/서버별 차별화된 운영적용이 가능하도록 한다.

□ 어플리케이션의 업무 특성

어플리케이션은 업무를 지원하기 때문에 업무 특성을 고려하여 운영관리 수준을 결정하는 것이 합리적이다. 어플리케이션의 업무 특성은 다음과 같이 분류될 수 있다.

- Cost plus Hard Benefit Driver : 조직의 핵심 가치사슬을 지원하여 조직에 이익을 가져오면서 비용을 소모하는 영업/매출 관련 트랜잭션 처리 업무, 또는 일반고객을 대상으로 하는 서비스 업무
- Cost Driver : 조직의 업무를 효율화하기 위하여 비용을 소모하는 경영관리 또는 일반 업무
- Soft Benefit Driver : 의사결정을 지원함으로써 간접적으로 이익을 가져올 수 있는 경영정보 분석/가공 업무

〈표 3-12〉 업무 특성 및 어플리케이션(예시)

어플리케이션 Type	비즈니스 특성	어플리케이션 예시
Type 1	Cost Plus Hard Benefit Driver	영업, 고객상담, 청구, 수/미납, 고객관리, 홈페이지 시스템 등
Type 2	Cost Driver	그룹웨어, 인사, 재무, 회계, 자산 시스템 등
Type 3	Soft Benefit Driver	EIS, DW, OLAP, Campaign Management, Data Mining 시스템 등

□ 서버의 IT 특성

서버는 어플리케이션을 개발, 테스트, 운영하는 플랫폼이기 때문에 서버 자체 용도와 서버가 지원하는 어플리케이션의 IT 특성을 고려하여 운영관리 수준을 결정하는 것이 합리적이다. 서버의 IT 특성은 다음과 같이 분류될 수 있다.

- 서버 용도 : 서버의 용도가 OA용, 개발용, 테스트용, 운영용으로 구분될 수 있다.
- Transaction 상의 특성 : 서버가 운영해야 하는 어플리케이션 특성상 사용자 접근 및 데이터 트랜젝션 면에서 Fluctuation 정도로 구분될 수 있다.
- 프로세싱 특성 : 서버가 운영해야 하는 어플리케이션 특성상 실시간(Real-Time) 프로세싱이 필요한 것인지, 배치(Batch) 프로세싱이 필요한 것인지를 나눌 수 있다.

〈표 3-13〉 IT 특성 및 서버 예시

서버 Type	IT 특성	서버 예시
Type 1	– 서버용도 : 운영용 – 트랜잭션 특성 : 사용자 접근 및 데이터×트랜잭션 면에서 Fluctuation 정도가 심한 어플리케이션 운영 서버 – 프로세싱 특성 : 실시간으로 처리해야 하는 어플리케이션 운영 서버	영업, Web 영업 운영서버
Type 2	– 서버용도 : 운영용 – 트랜잭션 특성 : 사용자 접근 및 데이터 트랜잭션 면에서 Fluctuation 정도가 심한 어플리케이션 운영 서버 – 프로세싱 특성 : 배치로 처리해야 하는 어플리케이션 운영 서버	수/미납, 청구 운영서버 재무, 회계 운영서버 OLAP 운영서버
Type 3	– 서버용도 : 운영용 – 트랜잭션 특성 : 사용자 접근 및 데이터 트랜잭션 면에서 Fluctuation 정도가 심하지 않은 어플리케이션 운영 서버 – 프로세싱 특성 : 실시간으로 처리해야 하는 어플리케이션 운영 서버	고객상담 운영서버 그룹웨어, 자산 운영서버 Campaign 관리, EIS 운영서버

Type 4	– 서버용도 : 운영용 – 트랜잭션 특성 : 사용자 접근 및 데이터 트랜잭션 면에서 Fluctuation 정도가 심하지 않은 어플리케이션 운영 서버 – 프로세싱 특성 : 배치로 처리해야 하는 어플리케이션 운영 서버	인사 운영서버
Type 5	– 서버용도 : OA용, 개발용, 테스트용 서버	개발용 서버 테스트용 서버

(나) 관리대상 선정 결과(예시)

〈표 3-14〉는 어플리케이션/서버에 대한 관리수준을 차등화하기 위하여 앞에서 설명한 어플리케이션의 업무 특성과 서버의 IT 특성을 고려하여 어플리케이션과 서버를 매핑한 결과이다.

〈표 3-14〉 업무/IT 특성에 따른 어플리케이션 및 서버(예시-1)

서버 Type \ 어플리케이션 Type		Type 1 Cost Plus Hard Benefit Driver	Type 2 Cost Driver	Type 3 Soft Benefit Driver
Type 1	– 운영용 – 높은 수준의 트랜잭션Fluctuation – 실시간 프로세싱	영업 , Web 영업 시스템 및 서버 〈GROUP 1〉		OLAP 시스템 및 서버
Type 2	– 운영용 – 높은 수준의 트랜잭션Fluctuation – 거의실시간/배치 프로세싱		재무, 회계 시스템 및 서버 〈GROUP 2〉	OLAP 시스템 및 서버
Type 3	– 운영용 – 낮은 수준의 트랜잭션Fluctuation – 실시간 프로세싱	고객상담 시스템 및 서버	그룹웨어, 자산 시스템 및 서버	Campaign Management, EIS 시스템 및 서버
Type 4	– 운영용 – 낮은 수준의 트랜잭션Fluctuation – 거의실시간/배치 프로세싱		인사 시스템 및 서버 〈GROUP 3〉	Data Mining 시스템 및 서버

Type 5	- OA용, 개발용, 테스트용 서버	영업, 고객상담, 청구, 수미납, 고객관리, Web 영업 등의 개발 시스템 및 서버	그룹웨어, 인사, 재무, 회계, 자산 등의 개발 시스템 및 서버	〈GROUP 4〉

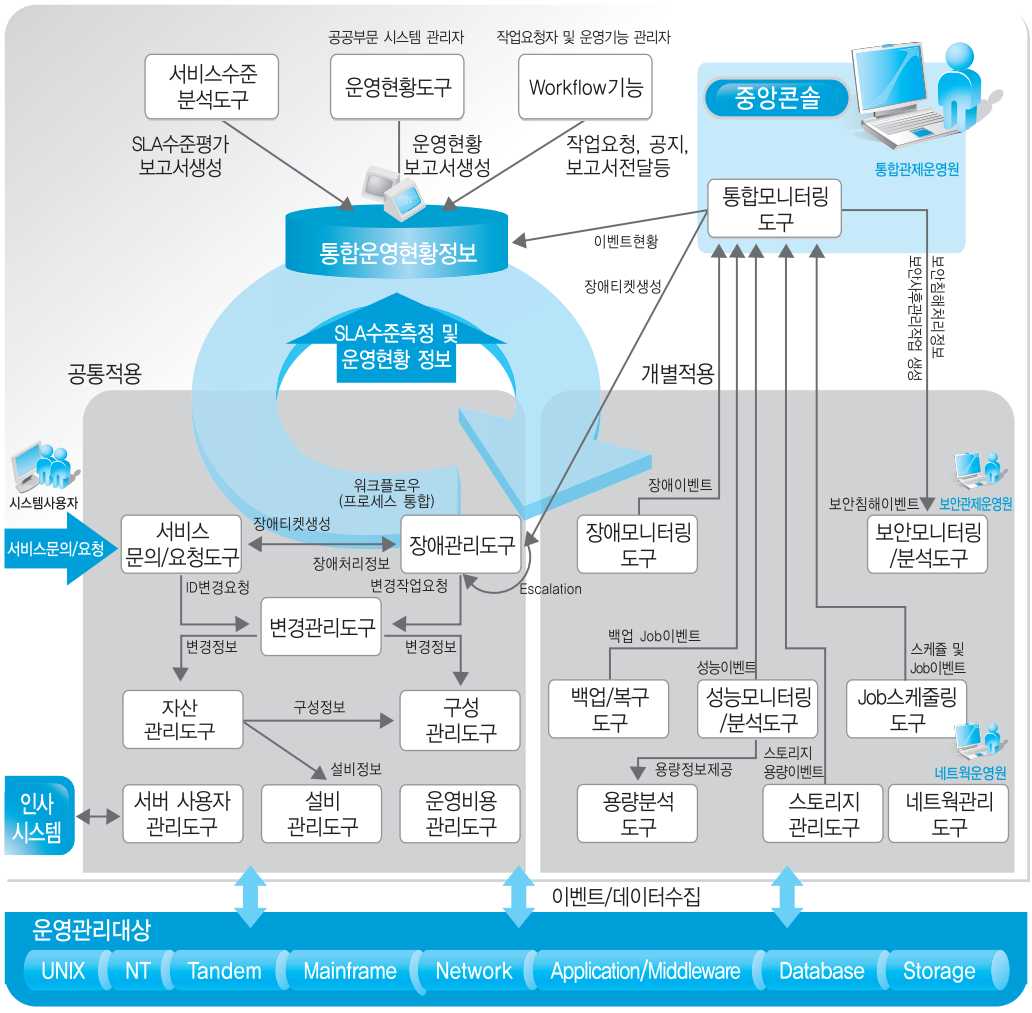
〈표 3-15〉는 업무 및 IT 특성을 고려하여 어플리케이션과 서버를 매핑한 결과(〈표 3-14〉 참조)에 따라 어플리케이션/서버의 관리 중요도를 결정하기 쉽게 하기 위해서 어플리케이션/서버 그룹별로 정리한 표이다. 즉, 〈표 3-15〉에서와 같이 어플리케이션의 비즈니스 특성과 서버의 IT 특성을 고려하여 선정된 결과에 따라 각 시스템별/서버별의 차별화된 통합적 성능관리를 적용하며, 운영관리가 되도록 한다.

〈표 3-15〉 업무/IT 특성에 따른 어플리케이션 및 서버(예시-2)

어플리케이션/서버그룹	비지니스/IT 특성	어플리케이션/서버예시
Group 1	- Cost Plus Hard Benefit Driver 또는 Cost Driver 어플리케이션 - 어플리케이션 상의 트랜잭션의 Fluctuation이 심한 서비스 지원 - 운영 환경의 시스템 및 서버	- 영업 , Web 영업 시스템 및 서버 - 수미납, 청구 시스템 및 서버
Group 2	-어플리케이션 상의 트랜잭션의 Fluctuation이 심한 서비스를 지원하거나 실시간 프로세싱을 지원	- 고객상담 시스템 및 서버 - 재무, 회계 시스템 및 서버 - OLAP 시스템 및 서버 - 그룹웨어, 자산 시스템 및 서버
Group 3	- Cost Driver 또는 Soft Benefit Driver 어플리케이션 - 어플리케이션 상의 트랜잭션의 Fluctuation이 심하지 않은 서비스 지원 - 거의실시간/배치 프로세싱 지원	- Campaign Management, EIS 시스템 및 서버 - 인사 시스템 및 서버 - Data Mining 시스템 및 서버
Group 4	- 개발용, 테스트용 시스템 및 서버	- 영업, 고객상담, 청구, 수미납, 고객관리, Web 영업 등의 개발 시스템 및 서버 - 그룹웨어, 인사, 재무, 회계, 자산 등의 개발 시스템 및 서버

다. 통합적 성능관리 구성도

성능관리 각 기능 및 기능간 연동을 자동화하는 통합적 성능관리 시스템의 구축을 통하여 업무 효율성 및 관리 효율성을 향상시킴으로써 궁극적으로 IT 서비스의 품질을 향상시킬 수 있다. (그림 3-12)는 통합적 성능관리를 중심으로 한 연관 부문과의 관계 및 흐름도를 종합적으로 표시해 놓은 통합적 성능관리의 논리적인 구성도이다.



(그림 3-12) 통합적 성능관리의 논리적 구성도

3 3 역할과 책임

가. 성능관리 책임자

성능 관련 협의된 서비스수준을 만족하도록 계획을 수립하고, 그 계획에 따른 성능 분석 및 개선을 수행하는 책임을 가지며, 다음과 같은 역할을 수행한다.

- 성능관리에 필요한 정책 및 관리 계획의 수립 및 변경, 통제 업무를 수행
- 성능관리 계획서를 수립하며, 각각의 기능별로 작성된 성능 보고서의 통합 및 통합적인 분석을 수행
- 성능 개선 방안 수립 및 검토, 개선 실행 등의 업무를 수행

나. 성능관리 담당자

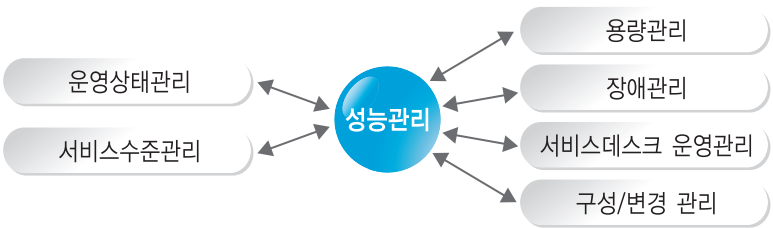
서버, 네트워크, 데이터베이스, 응용 소프트웨어 등의 성능 분석 대상별 성능 분석을 수행하여 문제점을 발견하고 해결 방안을 수립하는 책임을 지며, 다음과 같은 역할을 수행한다.

- 성능 분석 요구 사항 수집 및 성능 측정 계획 수립
- 관리항목별 성능목표 설정 및 성능측정 환경 구현
- 성능 모니터링 및 성능 데이터 수집, 보관
- 수집된 데이터로 임계치에 대한 사용 수준을 비교, 분석
- 계획된 업그레이드, 성능 튜닝을 위한 변경요청서 작성
- 성능 개선 방안 수립, 개선 실행 등의 업무를 수행
- 성능 측정 대상별 성능 추이 분석 및 보고서 작성

3 4 연계 분야

성능관리는 그 자체만으로 정보시스템 운영관리의 한 영역이 될 수도 있지만, 다른 관리 프로세스들과의 상호 연계를 통하여 효용성을 높일 수 있다. 운영상태관리를 통하여 모니터링된 정보시스템의 상태정보는 성능 관리 프로세스로 수용되며, 서비스수준관리를 통한 고객의 기대 수준은 성능관리 임계치에 반영된다. 또한 성능관리에 의하여 나타난 분석 결과들은 운영상태관리, 서비스수준관리, 용량관리, 장애관리, 서비스데스크 운영관리, 구성 및 변경관리 등에 적용되어 해당 관리 프로세스의 척도로서 적용된다.

따라서, 성능관리 책임자와 담당자는 성능관리의 연계분야에 대한 관계를 정확히 이해하고, 그 연관성에 의한 영향 관계를 정보시스템 운영관리에 반영하여야 한다.



(그림 3-13) 성능관리 연계 분야

3 4.1 운영상태관리와의 연계성

운영상태관리의 모니터링 과정에서 임계치를 초과하는 성능 측정 데이터가 관찰될 경우, 운영상태관리 담당자는 성능관리 담당자에게 분석을 요구(의뢰)할 수 있다. 시스템 구성요소에 대한 운영상태관리는 이상 징후를 발견, 기록, 분류, 통지하여 해당 업무 담당자를 통해 조치가 가능토록 함으로써, 시스템의 가용성을 향상시키는 업무이다. 따라서, 보다 적극적인 형태의 서비스 지원 및 안정화를 위하여 운영상태관리 분야는 성능관리 분야와 밀접한 지원관계를 유지하여야 한다. 운영상태관리는 협의된 서비스 수준에 따라 지속적인 운영 시스템 감시활동을 수행하게 되며, 감시활동에는 서비스에 영향을 줄 수 있는 징후들의 포착을 위한 일반적인 모니터링 업무 외에도

성능 감시와 장애 감시, 보안 감시활동 등이 포함될 수 있다. 성능관리 프로세스에서는 운영상태 관리 수행 동안 측정된 모니터링 데이터로부터 IT 시스템 성능 분석 요구사항 요건에 일치하는 관련 데이터를 받아들여 성능분석 자료로 활용한다. 운영상태 관리자는 분류된 데이터를 분석 후, 사전 협의된 세부 항목별 성능 임계치에 도달 또는 초과된 항목이 발견될 때와 기타 운영 간 비정상 성능 상태 발견 시에는 관련 프로세스 또는 성능관리 담당자에게 해당 내용을 통지한다.

또한, 성능관리 영역에서 측정 및 분석되어진 성능지표(임계치)는 운영상태관리의 임계치로 재반영되어 사용자의 만족도 및 SLA의 기준을 위한 기준치로서의 역할을 하게 된다.

3 4.2 서비스수준관리와의 연계성

서비스수준관리 구축 프로세스는 계획(Planning), 구축(Implementation), 운영(operation) 단계를 거쳐 수행된다. 성능관리는 구축과 운영 단계에 밀접한 관련이 있다. 성능관리를 통하여 측정 및 분석된 결과는 SLA의 기초자료로 활용되며 이 근거 자료를 이용하여 SLA가 설정될 수도 있다.

최종 SLA 완성본이 합의되는 즉시, 모니터링 기능과 이에 따른 문서화 작업이 수행되어야 한다. 서비스 품질에 대한 문서는 주기적으로 작성되어야 하며, 만약 SLA에 위반되는 평가결과가 도출 된다면, 별도로 이에 대한 문서를 작성하여 관리하는 것이 좋다. 작성되는 문서는 성능관리 담당자에게 전달 및 검토되어야 하며, SLA 검토회의를 통하여 서비스 품질 개선을 위한 조치가 이루어져야 한다.

이 운영단계에서는 SLA를 위반하는 이슈들이 성능관리 단계로 피드백(Feedback) 되며, 성능조정(Tuning) 프로세스를 거쳐 SLA 충족요건에 부합하도록 성능관리가 진행된다. 성능검토와 외부 계약당사자의 계약 내용에 따른 성과 평가 등을 위한 문서 역시 성능관리 프로세스를 통하여 산출된다.

사용자 만족도 조사를 통하여 서비스 성능평가와 함께 서비스를 제공받는 사용자 측의 서비스에 대한 만족도 역시 정기적으로 조사되어야 한다. 주로 사용자 만족도 조사방법은 설문에 의하며, 설문내용 및 시기, 대상 등은 서비스 제공자와 사용자의 협의에 따라 결정되는데, 이 결과 역

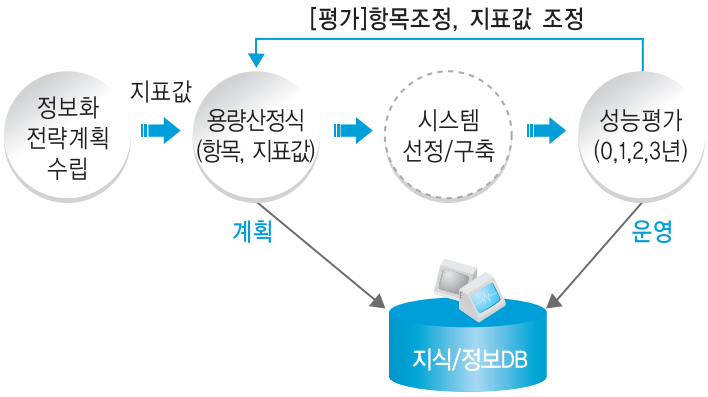
시 성능관리 영역으로 피드백(Feedback) 되어야 한다. 최근에는 외부 툴을 도입하여 설문조사 형태의 서비스 수준 협의 조사활동을 대체하는 경향이 있다. 클라이언트(Client)의 즉각적인 SLA 상황을 모니터링 하다가 SLA를 위반하는 사례가 발생하면 곧바로 분석되고, 평가되어 성능 분석 및 조정(튜닝) 단계로 연계시키고, 그 피드백을 되돌려 주는 솔루션을 사용함으로써, IT 인프라 성능관리 및 서비스수준관리를 자동화시키고 있다.

3 4.3 용량관리와의 연계성

하드웨어 용량산정에 있어서 적절한 용량산정 항목 및 정확한 지표 설정이 매우 중요한 것은 선정된 항목이나 지표 값에 따라 용량산정 결과가 큰 차이를 나타낼 수 있기 때문이다. 특히, 지표 값을 보다 정확하게 산출하는 방법이 핵심적인데, 이 때 성능관리(평가)를 통한 분석결과가 가장 정확한 기반이 될 수 있다. 일반적으로 용량산정 시에는 정보화 전략계획수립 결과나 구축할 정보시스템 분석설계서를 참고하며, 참고할 자료가 없는 경우에는 질의서를 이용한다. 질의서는 충분히 정의된 자료를 근거로 하지 않고 응답자의 경험과 개략적인 수치를 이용하기 때문에 용량산정의 정확성이 낮다. 그러므로 보다 정확한 용량산정을 위해서는 용량산정 항목과 지표에 관련된 내용을 포함하는 정보화 전략계획수립 보고서나 정보시스템 분석설계서로부터 도출된 정확한 성능분석(평가) 데이터를 활용하여야만 한다. 성능평가에 따른 하드웨어 용량산정식에서 각각의 용량산정 항목에 대한 검증이 필요한데, 이를 토대로 시스템을 선정하고 개발, 설치, 운영한다. 운영환경 하에서 성능평가를 하여 도입 시 예상한 성능지표 값과 차이를 분석하여 하드웨어 용량산정 시 적용된 항목과 지표 값의 타당성을 검증할 수 있을 것으로 생각된다.

성능평가에 따른 하드웨어 용량산정식과 더불어 지식/정보 DBMS 시스템 구축이 필요하다. 현재 추진하고 있는 정보화사업과 연계하여 각 사업에서 발생하는 지식을 데이터베이스화하고, 지식화 할 수 있는 시스템을 구축해야 한다. 정보시스템을 도입하는 기관의 정보화 전략계획수립 자료, 하드웨어 용량산정 관련 자료, 성능평가 항목과 성능평가 결과 지표, 차이분석 등의 결과를 데이터베이스화하여 용량산정식에 매우 중요한 항목과 지표 값 적용의 정확성을 높이는데 유용한 분석자료로 활용될 수 있게 한다.

위에서 설명한 하드웨어 용량산정과 관련된 정보화 전략계획수립, 용량산정식 도출, 성능평가, 지식/정보 데이터베이스화 간의 상호 연관관계를 도식화하면 (그림 3-14)와 같다.



(그림 3-14) 하드웨어 용량산정 과제 간 상호 연관관계

앞에서 설명한 용량산정과 관련된 정보화 전략계획수립, 용량산정식 도출, 성능평가, 지식/정보 데이터베이스화 과제를 전략적으로 수행하기 위하여 성능관리 담당자는 성능 분석 결과에 대한 데이터를 용량산정에 반영할 수 있도록 하여야 하며, 용량관리에 반영된 성능 분석결과에 대한 피드백을 다시 모니터링하여 평가한다.

3 4.4 장애관리와의 연계성

성능관리에서 보고된 성능 장애 현상은 장애관리로 이관되어 처리될 수 있다. 또한 장애관리상에서 발생하는 문제(Problem)에 대한 극복과정의 일환으로 다시 성능 조정이 요구되어질 수 있다. 장애관리는 응용 프로그램, 데이터베이스, 시스템, 네트워크 등 정보시스템 운영관리 대상 시스템의 고장, 장애, 서비스 불능 상태, 성능 저하 등 시스템의 장애 여부를 관찰, 진단, 보고, 제어, 처리하는 일련의 과정을 말한다. 성능 장애의 근본원인을 사전에 차단하기 위해 성능 문제의 발생 통계 및 관리를 통하여 성능 문제의 발생을 예측, 분석하여 중단 없는 서비스를 제공할 수 있도록 상호 보완적 업무 프로세스를 정립하여야 한다.

3 4.5 서비스데스크 운영관리와의 연계성

성능관리는 서비스데스크 운영관리와 밀접한 관계를 맺고 있다. 성능관리의 모니터링 단계에서 보고된 SLA 또는 성능 지표를 위반하는 결과는 서비스데스크 운영관리를 통하여 구성관리 데이터베이스(CMDB : Configuration Management Database)에 저장된다. 이 성능 장애 분석 결과는 운영 장애의 일종으로 판단하며, 다음과 같이 분류된다.

- 장애(Incident) : 장애(사고)는 정보기술 운영서비스에 영향을 주는 예상치 못한 사건
- 문제(Problem) : 문제는 단순한 사고가 원인이 되어 발생하지만 근본원인(Root Cause)을 파악할 수 없는 사건
- 알려진 오류(Known Error) : 알려진 오류는 문제에 대한 근본원인이 밝혀져서 향후 재발생 시 참조 가능한 상태의 사건

성능 장애는 그 분류 형태에 따라 서비스데스크를 통하여 장애관리, 문제관리, 또는 변경 및 구성관리 단계로 이관되며, 알려진 오류로 분류된 경우는 서비스데스크를 통하여 신속하게 조치된다.

운영장애와 설비장애 등이 직접적으로 정보시스템의 성능장애와 관련되는 경우(한 예로, 네트워크 장애의 경우 회선불량이나 트래픽의 과부하로 시스템의 성능이 저하되는 경우)가 있는데, 이러한 예는 일상적인 정보시스템을 관리하는 측면에서 자주 부딪히는 문제로써, 서비스데스크 운영관리를 통하여 신속하게 분석, 조치되어야 한다.

3 4.6 구성 및 변경관리와의 연계성

성능관리와 구성 및 변경관리 영역과의 직접적인 연관성은 미약하지만, 성능문제에 대한 해결책으로 하드웨어 부품의 교체, 운영체제의 업그레이드, 소프트웨어(응용 프로그램)의 수정, 패치(Patch) 등이 요구 된다면, 성능 분석 결과는 구성 및 변경관리 프로세스로 연계되어야 한다. 성능관리 사전준비 단계인 계획 단계에서 IT 자원의 정확한 구성 정보를 기반으로 성능관리 계획이

설립되어야만 하기 때문에, 구성 및 변경관리의 데이터베이스 정보가 반영되어야 한다.

또한, 성능 조정(튜닝) 단계에서 자원의 재배치나, 어플리케이션의 수정, 운영체제/DBMS의 파라미터 튜닝에 의한 성능 개선 효과가 없다고 판단될 경우, 용량관리에 의한 증설 작업이 수행되며, 이 증설 과정의 변경 사항은 구성 및 변경관리의 데이터베이스에 반드시 반영되어야만 한다.

3 5 용어정의

(1) 오버플로우(Overflow)

산술 연산 결과의 오버플로 때문에 생기는 오류. 이 오류 상태는 플래그 레지스터의 오버플로 플래그에 의해 표시된다.

(2) 충돌(Collision)

근거리 통신망에서 하나의 장치로부터 자료가 전송되고 있는 도중에 다른 장치에 의하여 자료가 전송되어 통신 매체 상에서 상호 간의 신호가 충돌하는 현상을 의미한다.

(3) 워크로드(Workload)

주어진 기간에 시스템에 의해 실행되어야 할 작업의 할당량을 의미한다.

(4) 쓰레드(Thread)

현재 작동 중인 한 태스크(작업)에서 작동할 수 있는 프로세스 또는 태스크보다 더 작은 하나의 작업 단위를 말하는 것으로 작은 서브 모듈을 말한다.

(5) 게이트웨이(Gateway)

통신 네트워크에서 서로 다른 네트워크들을 연결시켜 주는 장치를 말한다. 어떤 목적을 달성하기 위해 여러 유형의 네트워크들을 이용할 경우, 일반적으로 패킷 교환 네트워크를 경유하여 호출이 행해지는데, 한 네트워크에서 다른 네트워크로 보내지는 메시지나 포맷들은 모든 패킷 교환 네트워크에서 항상 동일한 것은 아니므로, 서로 다른 네트워크들을 연결시켜 주는 게이트웨이가 필요한 것이다.

(6) 스왑(Swap)

윈도우 내에서 가상 기억 장치를 구현할 때 사용되는 파일. 보조 기억 장치에 만들어져 있으며, 현재 사용되지 않는 파일을 보관할 목적으로 사용되며 마치 주 기억 장치의 일부분인 것처럼 사용된다.

(7) 데이터베이스(Database)

데이터베이스는 잘 정리된, 상호연관을 갖는 데이터의 집합이며 자료철이다. 어느 특정조직의 응용시스템들을 공동으로 사용하거나 또는 불특정 다수의 이용자들에게 유료로 정보 · 자료를 제공하기 위하여 컴퓨터가 접근할 수 있는 매체에 저장해 둔 데이터의 집합을 말한다.

(8) DBMS(Database Management System)

DBMS는 데이터베이스 관리시스템인 DataBase Management System의 약어로 데이터를 효과적으로 이용할 수 있도록 정리, 보관하기 위한 기본 소프트웨어를 말한다. DBMS는 데이터베이스를 관리하기 위해 필요한 수행과정인 데이터의 '추가', '변경', '삭제', '검색' 등의 기능을 집대성한 소프트웨어 패키지이며, 데이터베이스를 저장, 관리해야하는 기관이나 기업의 정보시스템 구축에 필요불가결한 소프트웨어로 주로 계층형과 네트워크형, 그리고 관계형(relational) DBMS가 존재한다.

(9) RDBMS(Relational Database Management System)

RDBMS는 열과 행으로 된 2차원의 표로 데이터를 표현하는 데이터베이스 관리시스템을 말한다. RDBMS는 관계 데이터베이스 모형(relational database model)에 따라 데이터베이스를 구성하고 관리하는 시스템이다. 기존에 보급되어 있는 UNIX나 PC를 플랫폼으로 한 데이터베이스 관리시스템의 대부분은 RDBMS이며, 기업내 정보시스템과 같이 데이터량이 많고 높은 신뢰성이 필요한 데이터베이스에 널리 이용되고 있다.

(10) 조정(튜닝)

튜닝은 조정이라는 뜻으로 컴퓨터 시스템 또는 소프트웨어의 효율성을 높이기 위하여 사용되는 일련의 개선 작업을 말한다. 일반적으로 성능의 척도는 실행 속도와 작업을 위하여 사용되

는 자원의 크기로 결정되며, 튜닝은 동일한 작업을 수행하면서 최소의 자원과 시간을 사용하여 작업을 수행할 수 있도록 취해지는 일련의 개선 작업이다.

(11) 트랜잭션(Transaction)

트랜잭션이란 컴퓨터 단말기의 사용자가 중앙처리 시스템에 요구하는 일(작업)의 단위로써 한번의 인터랙션(interaction)을 나타내는 말이다. 트랜잭션의 본래 의미는 '처리'로써, 구체적으로는 사용자의 요구에 의해 데이터가 일시적으로 시스템 내에 머물러 있는 것을 말하며, 파일이나 데이터베이스를 갱신처리(update)하는데 기본이 되는 하나의 입력 메시지를 뜻하기도 한다. 온라인시스템이나 시분할시스템에서 그 시스템의 단말기로 거래나 조회를 중앙처리시스템에 요구하는 것으로써 데이터파일 등의 마스터파일에 변경, 추가, 삭제 등을 하는 것을 말한다.

(12) 온라인 처리 프로세싱(OLTP : OnLine Transaction Processing)

OLTP는 네트워크상의 여러 이용자가 실시간으로 데이터베이스를 갱신하거나 조회하는 등의 단위작업을 처리하는 것으로 주로 신용카드 조회업무나 자동 현금 지급 등 금융 정보화 관련 부문에서 많이 이용하기 때문에 '온라인거래처리' 라고도 불린다.

(13) 데이터 웨어하우스(Data Warehouse)

데이터웨어하우스(Data Warehouse)란 기간 시스템의 데이터베이스에 축적된 데이터를 공통적인 형식으로 변환하여 일괄 관리하는 데이터베이스를 말한다. 데이터웨어하우스는 주요 의사 · 결정 · 자원 등 정보계 시스템에 사용하며 저장 및 분석, 방법론까지 포함하여 말하는 경우도 있다.

(14) 인덱스(INDEX)

인덱스란 색인 또는 지표 등의 뜻을 갖는 일반단어로, 정보통신 분야에서 인덱스는 색인 또는 지표의 뜻으로 파일이나 문서중의 특정내용의 위치를 알아내기 위한 키(key), 또는 참조를 위한 기호의 기능을 포함하고 있어 일반적으로 '표의 형식'으로 정의된다. 찾고자 하는 자료의 위치를 찾을 때 바로 이 색인(표)를 먼저 탐색하고, 참조기호에 의해 목적하는 데이터를 찾아

낼 수가 있기 때문에 처음부터 끝까지 모두 탐색하는 것보다 빨리 찾아낼 수가 있다. 책의 끝에 있는 색인과 같은 기능을 갖는다.

(15) 데이터베이스 성능 관리자(DPA : Database Performance Administrator

DPA는 데이터베이스의 성능관리를 전문적으로 담당하는 관리자를 의미한다. DPA는 일반 데이터베이스 관리자(DBA)들이 수행하는 업무의 상위에서 전반적인 DBMS의 성능이 업무 프로세스에 미치는 영향을 고려하며, 데이터베이스의 전략적인 활용도를 기준으로 최적의 성능을 유지시킨다. 중요한 데이터베이스 시스템의 경우 데이터베이스 성능관리를 위한 관리자(DPA)를 별도로 유지하는 것이 추세이다.

(16) 스키마(Schema)

스키마는 데이터베이스 시스템에서 DBMS에 의하여 관리되고 있는 대량의 자료들을 표현하기 위하여 사용되는 방법을 말한다. 또한, 스키마는 데이터베이스에 존재하는 자료의 구조 및 내용 그리고 이러한 자료들에 대한 논리적, 물리적 특성에 대한 정보들을 표현하는 데이터베이스의 논리적 구조를 지칭한다. 일반적으로 데이터베이스에서는 자료의 독립성을 위하여 여러 계층의 스키마를 사용하여 데이터베이스를 표현하고 있다.

(17) 정규화(Normalization)

데이터 처리에 관한 특정의 절차를 설정하는 것으로, 일반적으로 테이블 설계 시, 대상 엔티티(Entity)의 구체적인 속성을 고려하여 테이블을 분류하는 작업의 단계를 의미하며, 제1정규화(1NF), 제2정규화(2NF), 제3정규화(3NF), 제3정규화의 변형(BCNF), 제4정규화(4NF), 제5정규화(5NF)로 분류된다. 정규화는 깊어질수록 데이터 중복성이 줄어드는 반면 테이블 접근(Access) 속도는 느려진다. 따라서 일반적인 정규화는 속도 및 데이터 중복성의 Trade-off를 고려하여 제3정규화(3NF) 또는 BCNF 단계에서 결정된다.

(18) 병목현상(Bottleneck)

전체에서 다른 어떤 부분이 아무리 크다 해도 한 부분이 작게 되면 전체적으로는 작은 부분에 의하여 모든 행동이 결정되는 현상으로 최고의 부하 조건 동안에 혼선(traffic)을 줄이기 위하

여 제한되는 시스템 용량에 관한 용어를 말한다. 성능이 뛰어난 컴퓨터 시스템에서 어떤 하나의 부분에 대한 성능이 떨어진다고 하면 컴퓨터 시스템의 전체 성능은 바로 이러한 부분에 의하여 전체적으로 떨어지게 되며 컴퓨터 시스템의 작업의 흐름 역시 하나의 장치에 집중되어 이루어진다면 바로 이러한 작업에서의 성능이 전체적인 작업의 성능을 결정하게 된다.

(19) 로드 밸런싱(Load Balancing)

다중 처리 시스템에서 특정 처리기에 과중한 부하가 걸리지 않도록 시간 조정 또는 병렬 장비의 도입을 통하여 부하를 고루 분배하는 것을 의미한다. 미들웨어 서버, 또는 웹 서버, 네트워크 장비 등의 병렬화를 통한 하드웨어 레벨의 로드 밸런싱이 있고, 데몬(Daemon) 등 서비스 프로세스의 병렬 서비스를 통한 소프트웨어 레벨의 로드 밸런싱이 있다.

(20) 임계치(Threshold)

사전적 의미로는 '어떤 물리적인 현상이 다르게 나타나는 경계의 값'을 의미하며, 정보통신 부문에서는 장애상황 및 성능상태의 경계선으로써 의미를 가진다. 일반적으로 'Warning'과 'Critical'로 구분하여 설정하며, 기준치(Baseline)가 정상적인 상태에서의 표준상태를 의미하는 것과는 달리, 정상적 상황과 비정상적 상황의 경계를 의미한다.

(21) 저장 프로시저(Stored Procedure)

클라이언트/서버형 데이터베이스 시스템의 고속화 수법의 하나이다. 클라이언트부터 서버의 데이터베이스에 의뢰하는 명령(일반적으로 SQL문) 중 빈번하게 사용하는 일련의 명령군으로 데이터베이스 내부에 저장해 놓은 상태에서 사용한다.

(22) 공유 메모리(Shared Memory)

컴퓨터 프로그래밍에서, 공유 메모리는 일반 운영체계 서비스가 사용하는 읽기/쓰기 방식에 의한 것보다 프로그램 프로세스들이 데이터를 더 빠르게 교환할 수 있는 방법이다. 공유 메모리의 지정된 공간을 사용하면, 시스템 서비스를 사용하지 않고서도 데이터가 양쪽 프로세스에 의해 직접 액세스될 수 있도록 만들 수 있다. 공유 프로그램 코드 또는 데이터를 공유 메모리에 캐싱해 놓고 사용하게 되면 액세스 타임(Access Time)이 빨라지기 때문에 어플리케

이션의 효율이 개선된다.

(23) 세그먼트(Segment)

데이터베이스에서 세그먼트는 디스크 공간을 점유하고 있는 객체(Object)를 의미한다. 일반적으로 테이블, 인덱스 등이 해당되며, 정의(Definition)만을 저장하는 View, Stored Procedure, 시퀀스 등과 구별된다.

(24) 정렬 영역(Sort Area)

정렬 영역는 대상을 정렬하기 위해 사용하는 메모리 영역으로 데이터베이스 커넥션 형태에 따라 개별 프로세스의 세션 메모리에 설정되거나, 공유 메모리(Shared Memory) 영역에 설정되기도 한다. 데이터베이스의 group by, order by, union 등과 같은 정렬처리 명령이 수행될 때 이 정렬 영역을 사용한다.

(25) 롤백 세그먼트(Rollback Segment)

특정 테이블의 값이 바뀌게 되면(insert, update, delete), 해당 테이블의 변경 내용을 롤백 세그먼트(Rollback Segment)라는 디스크 공간에 기록해 둔다. 그리고 나중에 해당 테이블을 읽기 위해 접근할 때, 다른 트랜잭션에 의해 변경된 상태라면 현재 테이블의 값과 롤백 세그먼트에 기록된 변경 내용을 조합해 이전 테이블의 버전을 생성 해낸다.

(26) 배치(Batch)성

컴퓨터 시스템으로 데이터를 처리할 때 중단되거나 사용자와 대화하지 않고(Non-Interactive) 다른 작업들과 함께 연속적으로 실행되는 작업(Job)의 형태를 말한다. 일반적으로 OLTP성 작업(Job)의 상반개념으로 사용되기도 한다.

(27) 트리거(Trigger)

데이터베이스 내에서 참조 무결성을 유지하기 위해 특정 프로시저를 자동으로 호출하는 행동이다. 트리거는 사용자가 데이터를 삽입하거나 삭제하는 등과 같은 데이터 변경에 관한 시도를 했을 때 효력을 나타낸다. 트리거는 지정된 어떤 변경이 시도되면, 일련의 행동들을 취하도록 시스템에게 알릴 수 있다. 트리거는 부정확하고, 허가 받지 않았으며, 일관성이 없는 데

이터 변경을 방지함으로써 데이터베이스의 무결성을 유지하는데 도움을 준다.

(28) 파싱(Parsing)

컴파일러나 인터프리터가 원시 프로그램을 기계어로 번역하기 위해 프로그램을 문법적으로 해석하는 것을 말한다. 일반적으로 문법 (Syntax) 체크와 의미론(Semantics) 체크를 수행하여 정확성을 검증하는 절차를 의미한다.

(29) 단편화(Fragmentation)

단편화는 서로 다른 객체 레벨에서 일어나며 데이터베이스 또는 저장된 파일시스템의 성능에 악영향을 줄 수 있다. 스토리지 상에 저장된 단편화 파일 또는 데이터베이스의 세그먼트에 대해서는 재구성(Re-Organization) 작업 등을 통해서 단편화 방지 (De-Fragmentation) 처리를 하여야 한다. 이 작업은 성능관리에 매우 중요한 조정(튜닝) 방법 중 하나이다.

(30) 익스텐트(Extent)

디스크 등의 직접 접근 기억 장치에서 특정 파일이나 프로그램을 저장하기 위해 운영 체제 또는 DBMS에 의해 확보된 연속적인 기억 공간의 단위를 의미한다. 일반적으로 디스크 상의 트랙을 기준으로 연속공간이 할당되므로 접근(Access) 효율을 높일 수 있는 저장 기법(단위)이다.

(31) 경합(Contention)

① 다중 프로그래밍 시스템에서 여러 개의 태스크가 동시에 같은 파일이나 데이터베이스를 사용하려고 하는 것. 시스템 다운의 한 원인이다.

② 키제한된 컴퓨터 시스템의 자원(CPU, Memory, Disk I/O, Network I/O)을 사용하려는 프로그램의 집중현상을 의미한다. 일반적으로 자원에 대한 경합이 발생하는 것은 정상적이지만, 집중현상이 지나치게 되면 병목현상(Bottleneck)으로 나타나게 된다.

(32) 랫치(Latch)

DBMS의 주된 기능 중 하나인 동일 자원에 대해 동시 액세스를 관리하는 방법이다. 랫치는 매우 짧은 시간 내에 획득되고 해제된다. Queue를 통해 관리되지 않으므로 먼저 Request한 프로세스가 먼저 랫치를 획득한다는 보장이 없으며, 대부분의 경우 배타적(Exclusive) 모드로만

획득된다. 랫치는 주로 공유메모리의 특정 메모리 구조체에 대한 액세스(library cache latch, cache buffers chains latch) 혹은 메모리 할당 시(shared pool latch) 사용되거나 DBMS의 중요한 코드가 동시에 수행되지 않도록 하기 위한 용도로(redo writing latch) 사용된다.

(33) Enqueue

동일 자원에 대한 동시 액세스를 관리하는 방법 중 하나이다. Enqueue는 이름에서 보듯 Queue를 통해 관리된다. 대상 자원에 대한 Owner, Waiter, Converter Queue를 관리하면서 먼저 요청한 순서대로 Lock을 획득하도록 하는 구조이며, Exclusive 모드 뿐 아니라 다양한 수준의 공유를 허용한다. 대표적인 것이 테이블 데이터를 갱신할 때 사용되는 TM, TX enqueue이다.

(34) 세션(Session)

① 대화식 시분할 시스템에서 단말기 사용자가 시스템에 로그온하고 나서 로그오프하기까지 사이를 나타낸다.

② 데이터 통신에서 컴퓨터와 컴퓨터 또는 컴퓨터와 단말기가 접속을 유지하고 있는 기간을 나타낸다.

(35) 교착 상태(deadlock)

다중 프로그래밍 시스템에서 발견되는 현상으로 프로세스들이 서로 작업을 진행하지 못하고 영원히 대기 상태로 빠지게 되는 현상을 말한다. 교착상태가 발생하는 원인은 프로세스 사이에 할당된 자원의 충돌로 인하여 발생하게 된다. 따라서 어플리케이션 로직의 실수로 인하여 발생하는 경우가 대부분이다.

(36) 객체(object)

데이터베이스에서 객체는 테이블, 인덱스, 프로시져, 트리거, 뷰, 시퀀스 등 데이터베이스 내부에서 생성되어 관리되는 모든 대상을 의미한다. 데이터베이스 객체는 데이터와 이 데이터를 관리하려는 방법들이 분리되어 또는 통합되어 처리된다.

(37) 커서(Cursor)

데이터베이스 내부에서 데이터 처리 또는 SQL 코드의 처리를 위하여 임시로 사용하는 메모리 저장 영역이며, 공유 메모리나 혹은 프로세스 개별 메모리 영역에 위치한다. 커서(Cursor)는 데이터베이스의 저장 프로시저(Stored Procedure) 내부에서 사용되어지기도 하며, DBMS의 내부 처리 과정(Recursive Processing)에서 사용되기도 한다.

(38) 옵티마이져(Optimizer)

일반적으로 옵티마이져는 DBMS의 핵심 (소프트웨어) 엔진으로 SQL 문장의 내부 실행 계획(Execution Plan)을 생성한다. SQL 코드 자체는 실행 절차(Execution Procedure)를 포함하고 있는 것이 아니라, 처리해야할 데이터 집합의 범위만을 지정하고 있다. 따라서, 실행 계획(Execution Plan)은 옵티마이져에 의하여 데이터의 분포도, 처리범위, 처리비용(Cost) 등을 고려하여, 가장 효율적인 형태로 작성되어진다. 일반적으로 옵티마이저는 백그라운드에서 동작한다.

(39) ERD(Entity Relationship Diagram)

ERD는 개체관계도라고 하며, 언어로 기술된 사용자(대상자) 요구분석 사항을 그림으로 그려내어 그 관계를 도출하는 것을 의미한다. 조직 내의 대상(Entity)을 식별하여 프로그램의 소프트웨어적 대상으로 만드는 과정에서 식별되며, 객체(Object)를 추상적으로 기술하고 이들 간의 관계를 구조화한 것이다.

(40) 과부하(overhead) – 초과용량

컴퓨터에서 프로그램을 실행하는 데 수반되는 부수적인 작업 또는 그것에 소요되는 시간 즉, 프로그램의 관리, 스케줄링, 기억 장소 할당, 하드웨어의 제어 등 운영 체제에 의해 수행되는 부수적인 작업이나 또는 그것에 소요되는 시간을 의미한다. 사용자의 입장에서 보면 운영 체제가 자원을 이용하거나 동작하는 동안에는 프로그램이 실제로 실행되지 않으므로 낭비라고 간주된다. 따라서 오버헤드는 운영 체제를 평가하는 하나의 가늠이 되며, 작을수록 응답이 빨라진다.

(41) 병렬처리(Parallel Processing)

데이터베이스의 처리 속도를 향상시키는 수법의 하나. SQL 문을 분할해 여러 개의 CPU와 디스크를 사용해 동시 병렬로 처리를 실행한다. CPU나 디스크의 수가 늘어나면 반비례적으로 처리 시간이 짧아진다.

(42) 대역폭(Bandwidth)

① 초과될 수 없는 최상의 capacity

② Overhead를 무시한 측정치

(43) (실)처리량(Throughput)

① 특정시간 동안 실제 할 수 있는 작업의 량(what you really get)

② 대역폭 중 실제로 사용되는 capacity

③ 실제 처리량의 측정값은 다양한 요소들에 의해 영향 받는다.(H/W, S/W, human, Random access)

④ 정확한 처리량 값의 측정은 불가능하며, 단지 근사치(Approximated value)만을 구할 수 있다.

⑤ 최대 처리량을 측정한다면 100% 사용율(utilization)에서 얼마나 많은 작업량이 가능한가를 평가하는 것이 된다.

(44) 응답시간(Response Time)

사용자가 응답을 받기까지 걸린 총 시간으로, wait time과 run queue 대기시간이 포함된다.

(45) 서비스 타임(Service Time)

실제 request를 처리하는데만 사용된 시간(Queue 대기시간, Wait Time 제외) = actual processing time

(46) 사용률(Utilization)

측정대상 작업을 수행하기 위해 사용된 자원의 사용량(독자적이건, 종합적이건)으로, 백분율(%)로 표시된다. 일반적으로 busy%(률)로 나타낸다.

(47) 네트워크(Network)

통신서비스를 제공하기 위해 사용되는 회선, 네트워크 장비, 네트워크 S/W를 통칭한다.

(48) 네트워크 관리 시스템(NMS : Network Management System)

전사적으로 연결돼 있는 네트워크가 장애 없이 제대로 작동하고 있는지, 만일 네트워크에 부하가 걸려 있다면 그 이유는 무엇인지, 대역폭(bandwidth)의 사용은 무리가 없는지 등 네트워크 운영 전반에 관한 내용을 감독 및 관리하는 시스템이다.

(49) 전용회선

네트워크 서버, 통신장비, 단말기 등과 물리적으로 연결된 통신업체에서 제공하는 전용 네트워크 회선을 말한다.

(50) 전산시스템

업무를 전산적으로 처리하는데 사용되는 어플리케이션 프로그램과 상용화된 S/W 및 이들의 운영에 사용되는 H/W, 네트워크 등을 통칭한다.

(51) 전산자원

업무를 전산적으로 처리하는데 사용되는 H/W(서버, 네트워크 장비), S/W(어플리케이션 프로그램, 시스템 프로그램, 상용 솔루션) 및 부대시설을 통칭한다.

(52) 시스템

전산기와 전산기를 운영/관리하는데 사용되는 S/W를 의미한다.

(53) 장애

전산자원이 비정상적으로 작동되는 상황을 나타낸다.

(54) 침입방지 시스템(IPS : Intrusion Prevention System)

공격 시그니처를 찾아내 네트워크에 연결된 기기에서 수상한 활동이 이뤄지는지를 감시하며, 자동으로 모종의 조치를 취함으로써 그것을 중단시키는 보안 솔루션이다. 수동적인 방어개념의 방화벽이나 IDS와 달리 침입유도시스템이 지닌 지능적인 기능과 적극적으로 자동 대처하

는 능동적인 기능이 합쳐진 개념이다.

(55) 침입탐지 시스템(IDS : Intrusion Detection System)

방화벽과 함께 활용되는 네트워크 보안 솔루션으로 침입탐지시스템이 방화벽에 이은 차세대 보안 솔루션으로 부각되는 주된 이유는 방화벽이 해킹됐을 경우 이에 따른 피해를 최소화하고 네트워크 관리자 부재시에 시스템 자체적으로도 해킹 등에 대응할 수 있는 보안 솔루션에 대한 요구가 늘고 있는 상황에서 침입탐지시스템이 이같은 요구를 해결할 수 있는 솔루션이기 때문이다.

(56) 방화벽(Firewall)

방화벽은 인터넷과 같은 외부 통신 체제로부터 기업의 네트워크를 보호해주는 하드웨어 또는 소프트웨어 체제를 말한다.

(57) SLB(Server Load Balancing)

특정 서버가 오버로드가 되지 않도록, 즉 네트워크 서버간 로드 분산을 위하여 서버간의 트래픽을 효율적으로 분산하는 것이다.

(58) FLB(or FWLB : Firewall Load Balancing)

Layer3 Device로써 작동하며 방화벽으로의 각 연결(connection)은 IP Subnet별로 분리된다. FLB는 라우팅 기능으로 수행하며 FLB 결정에 Content 룰을 적용하지는 않는다.

(59) MTBF(Mean Time Between Failure)

인시던트가 복구된 시점부터 다음 인시던트가 보고된 시점 사이의 평균 시간, 즉 Uptime을 말한다.

(60) MTTR(Mean Time To Repair)

문제가 발생한 시점부터 복구한 시점까지의 평균 시간, 즉 Downtime을 말한다.

(61) ifInErrors

전송되어 들어온(inbound) 에러 패킷의 개수를 말한다.

(62) ifInUcastPkts

전송되어 들어온(inbound) 유티캐스트 패킷 개수를 말한다.

(63) ifInNUcastPkts

전송되어 들어온(inbound) 비-유티캐스트 패킷 개수를 말한다.

(64) ifInOctets

전송되어 들어온(inbound) 트래픽의 Octets 값을 말한다.

(65) ifOutOctets

전송되어 나간(outbound) 트래픽의 Octets 값을 말한다.

(66) IfSpeed

인터페이스(Interface)의 Speed 값을 말한다.

(67) 커널(Kernel)

운영체제(Operating System)에서 가장 핵심적인 역할인 자원(메모리, 프로세서 등)을 관리하며, 시스템이 원활히 돌아갈 수 있도록 제어해 준다.

(68) 트래픽(Traffic)

데이터의 흐름이 항상 일정하게 흐르는 것이 아니라 어느 시점에서 부하를 일으키는 것을 말한다. 통신장치나 시스템에 걸리는 부하를 말하기도 하며, 네트워크를 통해 움직이는 데이터의 양이나 어떤 종류의 트랜잭션 및 메시지 등의 양을 나타낼 때에도 사용된다.

(69) ping

현재 다른 시스템이 동작 중인지를 알아볼 때 쓰인다. ping의 기능은 IP 주소를 가진 시스템이 네트워크에 정상적으로 접속되어 있는지 여부와 내 시스템과 상대 시스템간의 지연상태와 품질상태를 알 수 있다.

(70) 네임서버(Name Server)

네임서버는 호스트의 어드레스를 영역 이름 형태로 주면 이에 대한 IP 어드레스를 구해서 넘겨주는 프로그램을 의미한다. 즉, 인터넷에서의 도메인 네임서버와 같은 의미로 도메인 네임을 IP 주소로 바꾸어 주는 역할을 한다.

(71) IP 주소

IP 주소는 TCP/IP 프로토콜을 사용하는 인터넷을 접속하는데 반드시 필요한 것으로써, 인터넷에 연결된 컴퓨터(기계)들을 식별하기 위해 32bit 숫자로 표현되는 고유 식별번호를 말한다.

(72) 패킷(Packet)

데이터 전송에서 사용되는 데이터의 묶음을 말한다. 각각의 패킷은 일정한 크기의 데이터뿐만 아니라 데이터 수신처, 주소 또는 제어 부호 등의 제어 정보까지 담고 있다.

(73) 원거리 통신망(WAN : Wide Area Networks)

광역통신망이라고도 불리는 원거리 통신망(WAN)은 근거리 통신망(LAN) 또는 중거리 통신망(MAN)을 다시 하나로 묶는 거대한 네트워크로써, 하나의 도시, 나라, 대륙과 같이 매우 넓은 지역에 설치된 컴퓨터들 간에 정보와 자원을 공유하기에 적합하도록 설계한 컴퓨터 통신망이다.

(74) 프레임 릴레이(Frame Relay)

공공 또는 사적인 전화선을 통해 데이터를 전송하는데 필요한 전송 기준이다. 데이터들은 각각 같은 크기로 쪼개져 프레임 안에 저장된 뒤 전송된다.

(75) 채널 서비스 장치(CSU : Channel Service Unit)

고속 디지털 전용 회선의 종단 장치이다. 장거리 통신 사업자들이 DS-0(64kbps), DS-1(1.544Mbps), DS-2(6.312Mbps) 등 고속 디지털 전용 회선의 물리적 특성을 관리하기 위해 원하는 고객의 구내에 설치하는 장치이다. 디지털 서비스 장치(DSU)나 다중화기(MUX)와 전용 회선 사이에서 인터페이스 및 버퍼로써 동작한다.

(76) 디지털 서비스 장치(DSU : Digital Service Unit)

디지털 회선용의 회선 종단 장치이다. 데이터 서비스 장치, 디지털 회선 종단 장치라고도 한다. 주 컴퓨터나 각종 데이터 단말장치(DTE)를 고속 디지털 전송로에 접속하여 데이터 통신을 하는데 필요하다.

(77) 브로드캐스트(broadcast)

다수의 국으로 동시에 정보를 보내는 것을 말한다. 동보 메시지는 다중점 회선상의 모든 국에 자발적으로 송신하는 메시지이다.

(78) 멀티캐스트(multicast)

구내 정보통신망(LAN)이나 인터넷에 접속되어 있는 일부 사용자 내에서 한 사람이 몇 사람에게 정보를 송신하고 그것을 수신한 몇 사람이 같은 내용을 버킷 릴레이(bucket relay)식으로 복수의 사람에게 송신함으로써 정보를 전파하는 특정 다수인에 대한 전송하는 것을 말한다.

(79) 영구가상회선(PVC : Permanent Virtual Circuit)

패킷 교환망이나 ATM 망에서 특정 단말기간을 고정적으로 접속해서 가상 회선을 설정하는 방식을 말한다. PVC 형태로 접속된 기기 간에는 언제나 데이터를 주고받을 수 있으므로 이용자는 전용 회선과 동일한 감각으로 이용할 수 있다.

(80) 가상 회선(VC : Virtual Circuit)

패킷 교환망에서 통신하려는 두 접속점 사이의 전송 경로가 논리적으로 고정되어 통신하는 동안은 물리적으로 고정된 전송 경로상에서 통신하는 것과 같은 효과를 나타내도록 하는 연결 방식이다.

(81) 역방향 명시적 혼잡 알림(BECN : Backward Explicit Congestion Notification)

프레임 중계 네트워크에서 통신 폭주 경로에 도달한 프레임의 반대 방향으로 전송되는 프레임에 사용되는 통지를 말한다. 역방향 명시적 폭주 통지(BECN)가 있는 프레임을 수신한 경우 필요하면 보다 높은 수준의 프로토콜에 의한 흐름 제어를 요청할 수 있다

(82) 순방향 명시적 혼잡 알림(FECN : Forward Explicit Congestion Notification)

프레임 중계 네트워크에서 수신 측에게 발신 장치에서 수신 장치까지 연결 경로에 통신 폭주가 발생했음을 알려주기 위해 사용되는 통지를 말한다.

(83) 간이 망 관리 프로토콜(SNMP : Simple Network Management Protocol)

가장 광범위하게 사용되고 있는 TCP/IP의 네트워크 관리 프로토콜이다. 원래 SNMP는 네트워크 관리를 위해 일시적으로 출현한 기술로 CMIP(Common Management Information Protocol)보다 기술적인 수준은 훨씬 낮은 프로토콜이다. 하지만 SNMP는 TCP/IP에 기반하고 있는데다 단순성과 오버헤드가 적다는 장점으로 인해 오히려 관리 프로토콜의 대세를 이루고 있다.

(84) 비동기 전송 모드(ATM : Asynchronous Transfer Mode)

디지털 데이터를 53 바이트의 셀 또는 패킷으로 나누어, 디지털 신호 기술을 사용한 매체를 통하여 전송하는 전용접속(dedicated-connection) 스위칭 기술이다.

(85) 트렁크(Trunk)

스위치 프레임에 수용되는 입출 회선에 설비되어 있는 장치를 말한다. 통화로의 일부를 구성하고 그 감시 제어를 하는 것으로, 통화에 필요한 전류 공급, 신호 송출, 요금 산출 기구 등을 갖는다.

(86) 오류 제어(error control)

전송 도중에 발생한 부호 오류를 검출하고, 정확한 정보를 재현하는 기술을 말한다. 오류 제어 기술로는 오류 정정 부호에 의하여 오류를 정정하는 순방향 오류 정정(FEC)과 오류 검출 부호를 사용하여 재송신하는 자동 재송 요구(ARQ) 등이 있다.

(87) 가상 패스 링크(VPL : Virtual Path Link)

비동기 전송 방식(ATM)에서 가상 패스 식별자(VPI) 값이 할당되는 점과 이 값이 해석되거나 제거되는 점 사이의 가상 채널 링크의 군(집합), 즉 하나의 공통 VPI에 의해 식별되는 가상 채널 링크의 군을 말한다. 가상 패스는 하나의 공통 식별자값에 의해 조합된 복수의 가상 채널에 속하는 ATM 셀들의 단방향 운반을 서술하기 위해 사용되는 개념이다.

(88) 가상 채널 식별자(VCI : Virtual Channel Identifier)

ATM 셀 서식에서 정의된 가상 채널(VC)을 식별하기 위한 16비트의 식별자이다.

(89) 관리 정보 베이스(MIB : Management Information Base)

망 기기를 감시, 제어하기 위해 사용되는 체계화된 관리 정보 항목을 말한다. 망에서 관리되는 장치가 정적 또는 동적 정보로서 유지되며, 관리 장치가 그 내용을 얻고 변경한다.

(90) 관리 정보 베이스 II (MIB-II : Management Information Base II)

TCP/IP 기반의 인터넷상에서 네트워크 관리 프로토콜로 사용하기 위한 MIB의 2번째 버전을 정의한다. 모든 SNMP 에이전트와 도구들은 MIB-II에 따르는 MIBs를 포함하며 모든 장치들은 MIB-II 표준을 따르는 최소한의 값들을 보여준다.

(91) 비차폐 연선(UTP : Unshielded Twisted Pair wire)

차폐 연선(STP)과는 달리 외부의 전계, 자계 또는 다른 전송선에서 유도되는 전계, 자계로부터의 영향을 차단하기 위해 도전성 물질이 많은 피복을 둘러싸지 않은 연선을 말한다.

(92) 라우터(Router)

근거리통신망(LAN)을 연결해주는 장치로써 정보에 담긴 수신처 주소를 읽고 가장 적절한 통신통로를 이용하여 다른 통신망으로 전송하는 장치이다.

(93) RMON(Remote network MONitoring)

원격지 통신망의 통신 정보를 수집하여 해석하기 위한 기능의 한 종류이다.

(94) 이더넷(Ethernet)

가장 대표적인 버스 구조 방식의 근거리통신망(LAN)이다.

(95) 반송파감지 다중접근/충돌검사(CSMA/CD : Carrier Sense Multiple Access with Collision Detection)

버스 혹은 트리 모양에 접속한 LAN(Local Area Network) 방식의 일종으로 미국 제록스 사

의 이더넷이 처음 사용하고, 그 후 미국 IEEE가 LAN의 표준 방식의 하나로 채용했다. 하나의 전송로를 다수의 사용자(단말)가 사용하기 위한 방식이다.. IEEE 802 위원회가 표준화한 LAN 액세스 방식에는 CSMA/ CD와 토큰 버스, 토큰링 세 가지가 있다.

(96) 주기적 덧붙임 검사(CRC : Cyclic Redundancy Check)

데이터 전송 과정에서 발생하는 오류를 검출하기 위하여 순환 2진 부호를 사용하는 방식이다.

(97) 토큰링(Token Ting)

토큰링 기법을 사용하는 네트워크를 말한다. 이 기법은 링 형태에 사용하기 위해 고안되었으나 지금은 모든 네트워크 통신망에 응용되고 있다. 토큰링은 완전한 분산 제어 방식으로 토큰이란 전기적인 표식을 이용하여 네트워크 사용을 결정한다. 네트워크 상에 오직 하나의 토큰만이 존재하며, 이 토큰을 가진 노드만이 네트워크 사용 권한을 가지고, 독점적으로 네트워크를 사용할 수 있다.

(98) 광섬유 분산 데이터 인터페이스(FDDI : Fiber Distributed Data Interface)

미국표준협회(ANSI)의 X3T 9.5 위원회가 1982년부터 표준화를 추진한 광섬유를 전송 매체로 하는 고리형 망에 컴퓨터나 단말 장치를 상호 접속하기 위한 인터페이스를 바탕으로 한 고속 구내 정보 통신망 (LAN)의 표준 규격이다.

(99) 인터넷 제어 메시지 프로토콜(ICMP : Internet Control Message Protocol)

TCP/IP 기반의 인터넷 통신 서비스에서 인터넷 프로토콜(IP)과 조합하여 통신 중에 발생하는 오류의 처리와 전송 경로의 변경 등을 위한 제어 메시지를 취급하는 무연결 전송(connectionless transmission)용 프로토콜(RFC. 792)이다. OSI 기본 참조 모델의 망 계층(Network Layer)에 해당된다.

(100) MAC 주소(MAC address)

일종의 고유번호로써 이더넷의 물리적인 주소를 말한다. 보통 랜카드의 주소라고 보면 되며 이 MAC 주소는 세계에서 유일한 번호이다.

4 성능관리 프로세스

성능관리 프로세스의 세부 단계별 내용들은 (그림 4-1)과 같이 정리할 수 있으며, 이 내용들은 분야(서버, 네트워크, DBMS, 응용 소프트웨어)별 특성을 반영하여 조정할 수 있다.

(그림 4-1) 성능관리 프로세스별 세부 내용

4 1 사전 준비

4 1.1 계획 수립 및 요구사항 검토

성능관리는 기존의 성능 모니터링 및 조정을 통해 IT 인프라 자원들이 최적의 조건으로 사용되고 합의된 성능 및 처리율이 달성되어 유지되도록 한다. 여기서 말하는 최적이란 합당한 비용으로 적재적소, 적절한 성능과 용량을 유지하는 것을 말한다. 성능 요구사항은 SLA에 합의되어 명시된 질적, 양적 기준에 따라 결정된다. 성능 요구사항에서 시스템의 운영과 업무 요구사항 사이에서 추구하는 목적의 차이를 극복하는 것이 중요하다. 일반적인 문제점은 IT 운영에 업무 관점이 배제된 전산 성능 측정치가 사용된다는 점이다. 서비스수준관리에서 IT서비스에 대한 업무 가치를 부여하고, IT조직은 그러한 서비스 요구를 만족시킬 수 있는 성능 계획을 수립하여야 한다. 이러한 관점에서 성능 계획을 수립할 때에는 사용자 요구사항을 수집하여 성능 관리 항목 및 목표를 설정하는데 반영하여야 한다. 또한 성능계획 수립을 위해서는 운영중인 시스템 구성 및 운영환경, 서비스 내용 등이 전반적으로 파악되어야 한다. 다음 내용들은 성능관리 계획 수립에 반드시 반영되어야 한다.

- 성능 병목현상 방지 및 해소 방안
- 용량계획 수립에 반영
- 서비스에 대한 관리보고
- 성능 모니터링 및 진단
- 작업 재분배(로드 밸런싱)
- 성능 추이 분석, 미래의 자원 요구 추정 자료 수집
- 서비스에 대한 처리량, 응답속도의 측정
- 성능 조정(튜닝) 및 장애 해결
- 문제가 발생하기 전에 미리 예측된 성능 문제를 예방

가. 현황 분석

성능관리 계획을 수립하기 전에 파악해야 할 첫 번째 요소로서 대상 IT 인프라 시스템에 대한 현황 분석을 들 수 있다. 시스템의 서비스 아키텍처, 업무 목표, 하드웨어 구성현황, 사용자들의 정보시스템을 사용하는 경향에 대한 분석이 선행되어야 성능관리 계획을 보다 현실적으로 수립할 수 있기 때문이다. 특히 '업무 관점에서 성능 요구사항'을 분석하는 것에서부터 성능계획 수립의 기준을 설립함으로써, SLA 및 사용자 만족도에 대한 현실적인 업무 특성을 고려한 임계치(Threshold) 및 기준치(Baseline) 설정이 가능해 진다. 업무 관점의 성능 요구사항은 다음과 같은 관점에서 고려되어야 한다.

- IT 투자에 대한 ROI의 최적화
- IT 서비스의 가용성 극대화
- IT자원 활용에 대한 효율성 제고
- 업무와 연계한 IT Planning 수립

나. SLA의 요구 성능 수준

SLA(서비스수준협약 : Service Level Agreement)란 서비스 제공자와 서비스 사용자가 제공될 서비스 및 그와 연관된 여러 조건들에 대한 서로의 책임과 의무사항을 기술해 놓은 협약서이다. 성능관리에 있어서 SLA는 서비스 사용자가 요구하는 응답시간(Response Time) 또는, 처리량 (Throughput), 가동률(Availability), 성능 장애율 등을 수치화하여 서비스 수준에 대한 명확한 기대치를 제공한다. 이 수치는 성능관리 계획에 반영되어, 성능상의 임계치 역할을 한다.

다. 사용자의 만족도 및 요구사항

성능관리 계획단계에서 가장 적극적으로 반영되어야 할 기준은 실질적으로 서비스를 사용하는 사용자의 만족도와 그들의 요구사항이다. 사용자(End-User)의 만족도는 최대한 수치화, 정량화하여 성능 상의 임계치로 환산하여 적용하거나, 조직의 업무 프로세스를 유지하기 위한 목표로

써 설정되어야만 한다. 예를 들어, 조직의 업무 목표를 달성하기 위하여 하루에 200만 개의 트랜잭션을 처리하여야 한다면, 이 각각의 트랜잭션에 대한 응답시간, 또는 처리량으로 환산되어 성능관리에 반영될 것이다. 또한 사용자가 수시로 느끼는 성능상의 문제사항은 수시로 성능관리 담당자에게 연계되어 성능 측정 항목으로 도입되고, 측정 및 분석되어야 한다.

라. 성능 평가 기준치(Baseline) 설정

서비스가 정상적으로 운영될 때의 기준치를 확보하는 것은 성능관리에 있어서 흔히 간과되기 쉬운 영역이다. 일반적으로 성능관리를 위한 모니터링은 성능상의 문제점이 발생한 기간에만 실시하는 경향이 있다. 그러나, 서비스가 정상적으로 운영될 때의 기준치를 미리 보유하게 되면, 성능상의 문제점이 대두된 기간과 비교 분석할 수 있게 되므로 성능 문제를 해결할 수 있는 매우 손쉬운 실마리를 제공받을 수 있게 된다. 성능 기준치(Performance Baseline)는 시스템이 정상적인 서비스를 제공하고 있는 시점에서 표준 업무시간대(예를 들어, OLTP 서비스의 09:00부터 18:00까지), 또는 업무 시간대 기준 사용량 최고치(Peak Time) 시간대를 기준으로 확보하였다가 성능상의 저하현상, 또는 장애가 발생한 경우 비교/분석하여 그 원인을 효율적으로 파악할 때 사용한다.

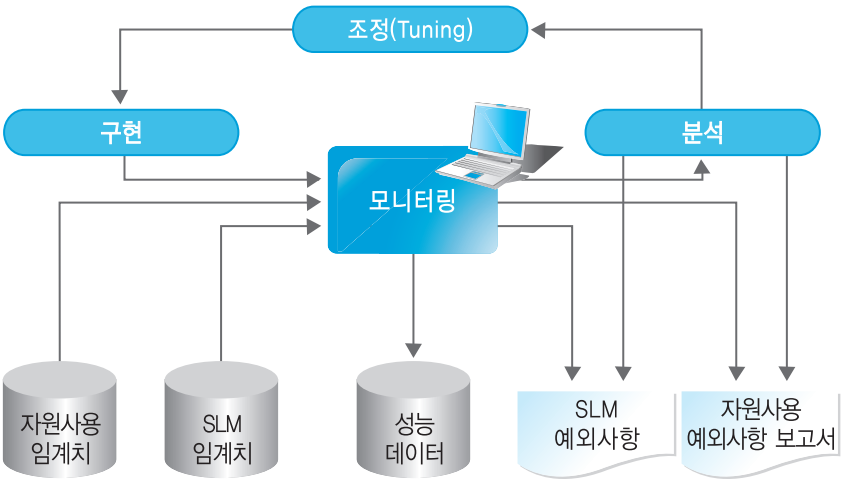
마. 기대효과

정보시스템의 성능에 대한 체계적인 관리는 계획 수립, 요구사항 검토, 성능 측정 및 분석 방안 수립, 성능 측정 환경 구현, 데이터 수집 및 분석, 성능 개선 방안 수립, 성능 조정 및 검증의 단계를 거쳐 진행되며, 이 각각의 단계는 구성 및 변경관리, 서비스수준관리, 운영상태관리, 서비스 데스크 운영관리, 장애관리, 용량관리 등의 연계분야와 효율적인 인터페이스를 통하여 운영된다. 따라서 명확한 성능관리는 정보시스템이 최적의 서비스 환경을 제공하고, 적절한 용량 계획의 기준을 제시하며, 시스템이 가지고 있는 자원을 최대한 적재적소에서 사용할 수 있도록 기여한다. 성능관리를 통하여 시스템 자원의 사용률(Utilization)과 응답시간, 처리량의 상관관계를 분석할 수 있게 되며, 따라서 불필요하게 비정상적으로 사용되는 자원의 병목현상을 분석하고, 조정(Tuning)할 수 있게 됨으로써 비용의 개선효과를 가져올 수도 있다.

4 1.2 성능 측정 및 분석 방안 수립

IT 인프라 시스템의 성능을 효과적으로 관리하려면 서버 자원의 사용방법, 기능별 사용량, 최대 사용 시간대 및 최소 사용 시간대를 식별하고 파악하여 제공되는 서비스가 관련 SLA 목표를 충족시킬 수 있도록 해야 한다. 서비스 성능 문제점을 미리 예상하는 것이 목적이지만, 자원 사용량 수준에 따라 서비스 품질이 달라 질 수 있다. 성능 관리자는 모니터링에 필요할 수 있는 도구에 드는 비용을 정당화하고 IT서비스의 원활한 실행을 보장하는 것이 중요하다. 일반적으로 모니터링 작업에는 일정한 기간이 소요되며, 장기간 나타나는 추세와 보고가 용량계획에 요약되어 나타날 수 있다.

성능관리는 성능 문제가 많이 밝혀질 수 있는 사전 대응 방식으로 수행되어야 한다. 성능 수집항목은 주로 사용량, 응답시간, 관련 자원들로 구성된 IT 서비스의 가용성이 모두 합쳐진 결과에 따라 결정된다. 모니터 활동은 모든 구성요소들과 각각의 서비스에서 수행되어야 한다. 수집된 데이터는 전문 시스템을 이용하여 분석함으로써 임계치(Threshold), 기준치(Baseline)에 대한 사용 수준을 결정할 수 있다. 분석된 결과는 보고서를 통해 기술되고, 이에 따라 적합한 권고사항을 만들 수 있다. 제시된 권고사항에 의해 성능관리자는 서비스를 균형적으로 재분배시키거나, 현재의 서비스 수준을 변경하거나, 자원을 추가 및 제거하는 활동을 수행한다. 그리고 다시 보다 나은 효과를 내도록 수행된 변경사항들이 모니터링되고, 구현된 수집방법에 따라 계속해서 데이터들을 수집한다. 이러한 성능과 관련된 데이터 수집, 분석, 조정 및 구현은 주기적이고 동일한 반복적 활동이 수행되어져야 하고, 이는 (그림 4-2)와 같이 일련의 사이클을 형성한다.



(그림 4-2) 성능관리의 반복적인 활동

가. 성능관리 대상 및 측정항목의 선정

측정 항목을 결정할 때에는 서비스 중인 각각의 IT 인프라 자원에 대하여 처리량, 응답시간, 자원 사용량, 효율성을 구분하여 데이터를 수집할 수 있도록 측정항목을 설정해야 한다. 또한 수집된 데이터를 바탕으로 총 자원 사용과 각각의 서비스가 특정자원에 부여하는 부하에 대한 세부 프로파일을 작성하여야 한다. 세부적인 측정 대상 및 항목은 분야별 성능관리 프로세스를 통하여 기술한다.

나. 측정 항목별 임계치

각 측정 항목의 임계치는 SLA를 만족하는 기준에 의하여 설정되며, 사용자의 만족도와 추가적인 요구사항을 고려하여 설정한다. 측정 세부 항목별 임계치는 IT 인프라의 개별 제작사(Vendor)에서 제공하는 가이드라인을 따르는 것이 일반적이지만, 업무 역할(Role)에 상응하는 공공기관의 활용 특성이 반영되어야 한다. 즉, 성능 임계치를 고정된 하나의 기준으로 설정하는 것보다, 업무 형태와 특성을 고려하여 사용자의 응답시간, 만족도 등을 적극적으로 반영하여야 한다.

▶ 임계치 설정시 고려 요소

- 제작사(Vendor)의 성능 판단 기준 자료 및 권고사항
- SLA 기준
- IT 인프라 업무 목적에 따른 활용 형태
- 사용자의 만족도(응답시간, 처리량) 평가 결과
- 운영상태 관리 분야로부터의 피드백

▶ 임계치의 유연한 관리

- 임계치는 고정적인 한 가지 기준으로만 유지되지 않으며, 다양한 환경적 요소를 반영하여 변경 및 조정 될수 있다.
- 임계치는 주변 요소(Server, Network, DBMS, Application)의 영향을 받아 변경될 수 있다.
- 성능관리 책임자는 연계 분야와의 인터페이스를 통하여 임계치를 유연하게 판단하여야 한다.

4 1.3 성능측정 환경 구현

가. 성능 측정 환경의 구현 요건

성능관리 항목과 측정지표로 등록되어 있는 사항들은 기본적으로 측정 가능해야 한다. 측정할 수 없는 사항들이 성능관리 측정항목에 포함되면, 서비스 평가 시 논쟁의 소지가 발생할 가능성이 크다. 이는 성능관리를 도입하는 효과를 반감시키며, 성능관리 정책의 신뢰도에 문제가 발생할 수 있다. 최근 정보시스템에 대한 모니터링을 손쉽게 수행할 수 있는 자동화 도구들이 다수 출시되어 있어, 이를 이용한 모니터링은 성능관리 서비스수준 측정에 도움을 줄 수 있다. 시스템에 대한 모니터링 이외에 서비스에 대한 사용자의 만족도, SLA 도입효과 및 자원 사용 임계치, 병목현상의 판단 등도 꾸준하게 모니터링 되어야 한다.

나. 성능 측정을 위한 자동화 도구 활용 방법

임계치와 비교하여 사용 수준을 모니터링하고 규칙에 따라 자동화된 동작을 수행할 수 있는 성능관리 자동화 도구들이 점점 포괄적이며, 지능적으로 되어 가고 있다. 이러한 자동화 도구는 전체 서비스에 참여하는 기본 구성요소들에 따라 개선된 정보 및 보기를 사용하여 기본 데이터가 표시되며, 몇몇 경우에는 상태표시기나 보기가 직접 업무 프로세스에 관련되기도 한다. 자동화된 도구를 활용하여 성능 측정을 적용할 때는 다음 사항들이 고려되어야 한다.

- 다양한 성능 데이터의 집계 및 데이터베이스의 저장을 통해 성능 추이 분석, 성능 계획, 시스템 용량 증설에 대한 예측 기능이 제공 되도록 자동화 도구를 설정한다.
- 실시간 또는 누적된 성능 정보를 다양한 테이블, 그래프 형태로 모니터링한다.
- 복수 시스템 및 복수 항목에 대한 성능 정보를 동일한 화면에서 모니터링한다.
- 모니터링된 성능 측정치에 대해 실시간으로 산술 연산 및 논리 연산을 통해 가공된 성능 정보를 제공할 수 있도록 자동화 도구를 설정한다.
- 사용자 요구에 따라 각 관리대상 서버의 상황에 맞게 관리항목을 추가, 수정, 삭제한다.
- 각 대상 시스템에 대한 성능항목 및 측정 주기, 측정 방법, 저장방법을 적용한다.
- 성능 데이터의 상대적 변동량에 대한 임계치를 설정하여 임계치 초과 이벤트 정보는 성능 관리자에게 전달될 수 있도록 한다.
- 임계치 초과 이벤트 정보는 단문메시지, e-mail, 경보음 등 자동 통보 기능을 이용한다.

외부 툴 및 프로그램으로 구현된 모니터링 및 분석 툴은 운영상태관리 및 서비스데스크 관리 등의 1차 지원 관리 시스템과 밀접한 관련을 맺고 성능관리와 인터페이스를 유지하여야 하며, 성능분석 대상인 경우는 장애관리와 용량관리 영역에 분석된 정보를 자동으로 이관할 수 있는 기능을 보유하여야 한다.

다. 성능 측정의 영향도 분석

성능 측정을 위하여 설정된 측정환경이 오히려 시스템의 과부하 (OverHead)로 작용하여 전체적인 IT 인프라 시스템의 성능 상태를 저하시키는 사례가 종종 발생한다. 예를 들어, 과도하게 확장된 세그먼트 정보가 자료사전에 저장되어 이 정보를 쿼리하는 동안 DBMS 내부 엔진의 속도가 저하되는 경우 등을 들 수 있다. 또한 외부 툴 및 프로그램의 도입이 시스템 자원(CPU, 메모리, 디스크 I/O)을 잠식하여 정상적인 서비스 어플리케이션이 사용하여야 할 자원을 빼앗는 경우도 발생할 수 있다. 모니터링 및 분석 작업 중, 어쩔 수 없이 시스템 과부하를 발생시킬 수밖에 없는 작업이라면 Peak Time을 피하여 실행하여야 한다.

라. 성능 측정 항목별 성능 추이 및 성능 수준을 분석

- 개별적인 측정 항목별 성능 데이터의 기록을 누적한다.
- 누적된 개별 데이터는 추이 분석을 통하여 일별, 주간별, 월별 성능 상태를 분석한다.
- 각각의 측정 항목별 분석 기록을 기반으로 DBMS 전반의 통합적인 성능분석을 위하여 종합 분석한다.

마. 성능 측정 결과를 이용한 성능 분석 주기

- 성능 측정 결과 안정된 성능 상태를 보이는 측정항목에 대하여는 성능 분석 주기를 조절할 수 있다. 일간주기로 측정되던 항목이 수개월간의 분석 결과 안정화된 성능 상태를 지속적으로 보이는 경우, 이 항목의 측정 주기는 주간 또는 월간 단위로 변경할 수 있다.
- 일간주기로 실시하는 분석 방법은 실시간 또는 5~10분 이하의 짧은 간격으로 측정하여 즉각적인 성능상태를 모니터링하고 분석할 수 있는 데이터 수집을 목적으로 한다. 이 경우 데이터의 수집이 시스템의 과부하(Overhead)로서 작용할 가능성이 있으므로 주의하여야 하며, 만일 이러한 성능 데이터 수집 항목이 안정화된 성능상태를 지속적으로 나타내거나, 성능상태가 변경될 가능성이 거의 없을 경우는 주간, 또는 월간 주기로 변경할 수 있다.

바. 종합적인 관점에서의 성능 추이 분석 보고서를 작성, 기록 및 유지

성능 측정을 통하여 분석된 각 구성요소별 데이터는 추후 통합성능관리 분야에서 취합되어 업무 서비스 관점의 종합 성능 분석을 위하여 사용된다.

4 2 분야별 성능관리 프로세스

4 2.1 서버

가. 데이터 수집 및 분석

(1) 측정 항목

측정 항목을 결정할 때에는 서비스 중인 각각의 서버에 대하여 처리량, 응답시간으로 구분하여 데이터를 수집할 수 있도록 해야 한다. 또한 수집된 데이터를 바탕으로 총 자원 사용과 각각의 서비스가 각각의 특정자원에 부여하는 부하에 대한 세부 프로파일을 작성할 수 있다.

서버 성능관리를 위한 구성요소별 주요 측정항목은 다음 표와 같다.

〈표 4-1〉 CPU 측정항목

측정항목	설명	측정항목
총 CPU 사용율(%)	- CPU가 idle하지 않았던 시간의 비율(%) - 시스템모드 사용율(%) + 사용자모드 사용율(%) - 총 CPU사용율 + idle사용율 = 100%	실시간
시스템 모드 사용율(%)	- 시스템 모드에서의 CPU 사용율(%)	실시간
사용자 모드 사용율(%)	- 사용자 모드에서의 CPU 사용율(%)	실시간
Run Queue	- 수행중이거나 수행 대기중인 프로세스(runnable processes)의 평균 개수 - run queue가 클수록 CPU의 서비스를 받기 위해 대기하고 있는 프로세스의 개수가 많다는 것을 의미함	실시간

Pri Queue	– Priority에 의해 수행되지 못하고 있는 프로세스의 평균 갯수	실시간
Active 프로세스 수	– CPU를 사용한 프로세스의 갯수	실시간
사용자 수	– 로그인한 사용자 수	

〈표 4-2〉 메모리 측정항목

측정항목	설명	측정항목
총 메모리 사용율(%)	– 서버 Physical 메모리의 사용율(%) : 시스템 메모리(Kernel이 사용하는 메모리), 버퍼 캐쉬, 사용자 메모리가 포함	실시간
시스템 및 버퍼 캐쉬(%)	– 시스템(Kernel)과 버퍼 캐쉬(Buffer Cache)의 메모리 사용율	실시간
Page Request Rate(초당)	– 디스크를 통한 Page요청의 수	실시간
PageOut Rate(초당)	– Virtual 메모리로의 Page Out 수	실시간
Swap Out Rate(초당)	– Deactivation	실시간
메모리 Queue	– 메모리에 의해 수행되지 못하고 있는(가상 메모리 Access를 위한 Queue) 프로세스	실시간
Swap 공간 사용율(%)	– Running 프로세스에 의해 Reserve된 Swap 공간	실시간
메모리 캐쉬 Hit PCT(%)	– 찾고자 하는 file 시스템 내의 데이터가 메모리 버퍼 캐쉬에 있을 비율	실시간

〈표 4-3〉 디스크 측정항목

측정항목	설명	측정항목
Peak 디스크 사용율(%)	- 사용율이 가장 높은 디스크의 사용율(%)	실시간
DISK IO Busy(%)	- 디스크의 Read/Write 속도(KB/s) - 초당 디스크로 또는 디스크로부터 전송되는 데이터의 평균 KB수 - 모든 형태의 Physical IO가 포함됨	실시간
Raw IO Rate(초당)	- raw read와 write의 초당 횟수	실시간
디스크 queue	- 디스크 서브시스템에 의해 block된 프로세스의 평균 수	실시간
디스크 수	- 디스크의 수	실시간

〈표 4-4〉 프로세스 측정항목

측정항목	설명	측정항목
프로세스 부하	- 시스템을 집중적으로 사용하는 프로세스	실시간
Zombie 프로세스	- 아무런 수행작업 없이 존재하고 있는 프로세스	실시간

〈표 4-5〉 커널 측정항목

측정항목	설명	측정항목
파라미터 셋 확인	- 파라미터 셋 설정	비실시간

〈표 4-6〉 파일시스템 측정항목

측정항목	설명	측정항목
파일시스템 Rate(초당)	- 파일시스템 디스크 Physical read와 write의 초당 횟수	실시간
Peak 파일시스템 공간사용율 (%)	- 파일시스템 공간 사용율이 가장 많은 디스크의 공간 사용율	실시간

〈표 4-7〉 네트워크 I/O 측정항목

측정항목	설명	측정항목
네트워크 Packet율(%)	- 초당 발생하는 모든 인터페이스에 대한 성공적인 패킷(에러나 Collision이 없이 처리된 inbound와 outbound 패킷)의 수	실시간
네트워크 Collision율(%)	- 총 outbound패킷에 대한 Collision의 비율(%)	실시간
에러율(분당)	- 네트워크 인터페이스상의 에러 수	실시간
네트워크 트래픽	- 네트워크 인터페이스별 트래픽량	실시간

(2) 측정 방법

성능 모니터링을 위해 각각의 서버와 서버의 구성요소들에 대해 성능 측정이 가능한 환경을 구성한다. 서버의 구성요소에 대한 측정을 위해 OS에서 기본으로 제공하는 명령어를 이용하거나 시스템 성능 측정 도구를 설치하여 자원의 사용현황 및 응답속도, 병목현상에 대하여 데이터를 수집한다. 경향 분석을 위해 측정 항목별로 일정기간 혹은 주기적으로 데이터를 수집하여 보관한다.

시스템 성능 측정 자동화 도구를 사용할 경우 주기적으로 이벤트를 발생시켜 나타나는 서버의 성능을 측정하고 데이터를 보관한다. 또한 성능분석으로 인하여 발생할 수도 있는 시스템 성능 저하에 대하여 사전에 고려되어야 한다.

(가) 측정 주기

서버의 성능 데이터를 각 수집 항목별로 측정 주기에 맞추어, 24×7일간 중단 없이 측정한다. 단, 측정 주기는 서버의 중요도, 분석방법에 따라 설정하도록 한다.

(나) 수집된 데이터의 보관

시스템 성능에 대한 경향 분석, 향후 시스템 사용예측을 위한 데이터로 활용하기 위해서는 측정된 성능 데이터를 일정기간 보관하도록 한다. 데이터의 보관 장소는 향후 분석을 용이하게 할 수 있도록 스프레드쉬트를 이용하거나 데이터베이스에 저장, 보관하도록 한다.

(다) 측정 방법

서버의 성능 분석은 성능분석 지원도구를 이용하거나 OS에서 제공하는 모니터링 도구를 이용하여 CPU 및 메모리 사용, 스왑 비율, 디스크 I/O 비율 및 대기 시간 등을 포함한 시스템의 데이터 영역을 다각도로 측정하여야 한다. 각 항목들 별로 권장하는 수치(기준치)로 성능을 평가하고, 이 데이터를 종합적으로 분석하여 현재의 하드웨어 구성으로 서버 성능을 향상시킬 수 있는 방안을 모색하며 시스템 자원을 추가해야 하는 부분을 파악한다. 분석을 통해 작업 유형에 따른 자원 소비량과 전체적인 자원 로딩 비율을 파악할 수 있다. 따라서 서버의 성능 분석은 고성능 IT환경을 위한 첫 번째 수행 단계이며 보다 정밀한 용량계획 분석을 위한 사전 작업이기도 하다.

〈표 4-8〉 성능 데이터 수집을 위해 OS에서 제공하는 명령어들

명령어	기능 설명
sar	- CPU, 실제 메모리 사용, Paging, Swap, kernel memory, 프로세스 상태, file system 등의 데이터를 측정하는데 사용 - 데이터를 특정파일에 저장하여 기록하고 필요시 볼 수 있도록 한다.
vmstat	- 주로 프로세스와 관련된 측정 자료 (CPU, 메모리) - 기능 • CPU : Process, interrupt, CPU cache, and CPU activity • 메모리 : Virtual memory and swapping activity • I/O : Disk activity
iostat	- I/O Device에 대한 성능 데이터를 제공 - 기능 • CPU 이용도 및 TTY activity • 디스크 activity와 device errors • 초당 Mbyte, 디바이스별, 파티션별 데이터 제공
netstat	- 네트워크 관련 Activity에 대한 성능 데이터 및 다른 네트워크 인터페이스의 세밀한 정보를 제공 - 기능 • Soket state, interface state, DHCP information, arp, routing tables, streams state - 각 인터페이스에 대해 주고받는 패킷의 수 - 스택 정보 (tcp segment 측정, UDP, 다른 프로토콜 메시지) Overflow 발생 문제를 확인

(3) 분석 방법

모니터링으로부터 수집된 데이터는 정규 사용, SLA, 또는 수립되어질 수 있는 기준치(baseline)등에 대한 추세를 판단하기 위해 된다. 이러한 기준치와 수집된 데이터를 비교하고 정기적으로 모니터링 함으로써, 개별 구성요소를 사용하는데 있어 예외조건 또는 서비스 임계치가 정의될 수 있고, SLA내의 위반 또는 누락현상이 보고될 수 있다. 또한, 이러한 데이터는 향후 자원 사용량을 예측하는데 사용된다.

데이터의 분석을 통해 다음과 같은 문제점들을 식별할 수 있다

- 경합(Contention) : 데이터, 파일, 메모리, 프로세서
- 사용가능한 자원에 대한 부적절한 워크로드(Workload)분배
- 부족한 자원 : CPU, 메모리, 디스크 등 충분하지 않은 구성요소들
- 메모리의 비효율적인 사용
- I/O의 조각화 현상
- 어플리케이션, DBMS 설계에 있어서 비효율성
- 트랜잭션 비율의 예상치 못한 증가

(가) 측정항목에 대한 기준치 설정

요구사항 검토 및 서비스 특성, 과거 수집한 데이터를 분석하여 서버 성능 평가를 위한 임계치(threshold)와 기준치(baseline)를 식별하고 설정한다. 신규로 도입된 시스템 또는 과거 성능 데이터 부족으로 기준치를 설정하기 곤란한 시스템에 대해서는 공급자가 제시하는 일반적인 서버 성능 기준치를 우선 반영하고 주기적인 성능 측정을 통해 데이터를 분석하여 기준치를 설정하도록 한다.

서버 성능 기준치는 각 기관의 업무특성에 따라 다르게 설정하여야 하며, 기준치 설정은 다음을 고려하여 정한다.

- 요구사항 검토 결과
- 각각의 서버가 제공하는 서비스의 특성(온라인, 배치 등)
- 서비스의 Peak 시간대, 사용자 수 등
- 과거 서버의 성능 데이터 분석을 통해 얻어진 결과치
- 시스템에 설치되어 운영되는 어플리케이션의 특성

〈표 4-9〉 일반적인 서버 성능 기준치(예시)

수집 항목	기준치 예시	허용치 범위
o CPU - 총 CPU 사용율(%) - Run Queue Size	- Peak 시간 CPU사용율 80% 이하 - CPU당 평균 3개 이하	+5%
o 메모리 - 총 메모리 사용율(%) - Page Out Rate - Swap Out Rate	- 메모리 사용율 80% 이하 - 평균 Page Out ≒ 0 - Swap 사용율 20% 이하	+5%
o 디스크 - 디스크 사용율	- 디스크 사용율 80% 이하	+5%

(나) 분석 주기

성능 분석 및 조정은 성능 분석 주기에 따라 그 방법이 달라 질 수 있으므로 각 자원과 서비스의 사용 분석은 기록된 기간 동안의 단기, 중기, 장기, 최소, 최대, 평균 사용에 대해서 고려되어야 한다. 일반적으로 단기패턴은 24시간 동안 사용율을 다루고 중기패턴은 1주에서 4주, 장

기패턴은 1년 동안의 사용율을 분석한다.

이러한 각각의 기간에 대하여 사용량을 이해해야 하는 중요한 이유는 서비스 사용량의 변동이 개별 자원의 예상 변동 사용수준과 관련될 수 있기 때문이다.

〈표 4-10〉 주기별 성능분석의 종류

분석종류	내용
일간 성능분석	각 측정항목별 시간대별 성능 추이 및 특정 경향을 벗어나는 성능 요소를 분석한다.
주간 성능분석	업무 특성상 주간 단위의 성능 추이 분석이 필요할 때 실시한다.
월간(분기) 성능분석	일간, 주간 데이터를 취합하여 해당 월간(분기) 성능추이를 분석하고 지나간 월간(분기) 성능분석 자료와 비교하여 성능 권고안을 작성한다.
연간 성능분석	서버 용량에 대한 개선안 및 월간(분기) 단위로 작성된 개선안을 요약한 자료를 작성한다. 단, 작성 시기는 업무계획이나 예산 주기를 고려하여 정한다
비정규적인 성능 분석	SLA를 위반하는 현상이 발생하거나, 사용자의 요구가 있다면, 특정 IT서비스에 대해 비정규적인 성능 측정 및 분석을 실시한다. 특정 IT서비스에 영향을 주는 특정 하드웨어나 소프트웨어 자원을 식별하는 능력은 정확하고, 종합적인 성능분석 자료를 통해서 가능하다.

(다) 분석 방법

시스템 성능분석은 성능분석 자동화 도구를 이용하거나, 성능관리자에 의해 OS에서 제공하는 모니터링 도구를 이용하여 CPU, 메모리, 디스크 I/O, 네트워크 디바이스 등에 대한 수집된 성능측정 데이터를 각 항목별로 권장하는 기준치에 대한 성능과 비교 분석한다.

서버의 주요 성능 관리항목에 대한 분석방법은 다음과 같다. 여기서 제시하는 기준치는 예시이며, 각 기관에서 설정된 해당 측정항목에 대한 기준치에 의해 비교되어야 한다.

□ CPU 성능분석

성능 자동화 도구나 OS 기본 명령어인 vmstat나 sar 명령어를 이용하여 시스템의 CPU 자원 사용현황을 분석한다.

〈표 4-11〉 CPU 성능분석

분석 항목	내용	기준치(예시)
CPU 사용율(%)	- CPU가 Idle하지 않았던 시간의 비율(%) - 총 CPU사용율(%) = Sys사용율(%) + User 사용율(%) + WIO(%) - 총 CPU사용율(%) + Idle비율(%) = 100%	Peak 시간 CPU 사용율 80% 이하
Run Queue Size	- 수행 중이거나 수행 대기 중인 프로세스 (Runnable Process)의 평균 개수 - Run Queue Size가 클수록 CPU의 서비스를 받기 위해 대기하고 있는 프로세스의 개수가 많다는 것을 의미함	CPU당 평균 3개 이하

□ 메모리 성능분석

성능 자동화 도구나 OS 기본 명령어인 vmstat 명령어를 이용하여 시스템의 메모리 자원 사용현황을 분석한다.

〈표 4-12〉 메모리 성능분석

분석 항목	내용	기준치(예시)
Physical Memory 사용량	- 서버의 Physical 메모리 사용량 - 시스템 메모리(OS 사용영역), 파일시스템 블록 버퍼, 사용자 어플리케이션 메모리 등이 포함됨	Page Out이 발생하지 않을 정도의 Physical memory 여유율 확보
Swap 사용율(%)	- Swap 사용율	Swap 사용율 20% 이하

□ 디스크 성능분석

성능 자동화 도구나 OS 기본 명령어인 iostat 명령어를 이용하여 시스템의 Disk I/O 현황을 분석한다.

〈표 4-13〉 디스크 성능분석

분석 항목	내용	기준치(예시)
Disk Busy(%)	- I/O 활동으로 인하여 해당 디바이스가 사용이 활발했던 시간의 %	Disk Busy(%) 20% 이하
Disk Usage(%)	- 디스크 장치별 사용율	사용율 80% 이하 (업무 증가량 및 신규 추가 업무량을 수용할 수 있는 적정 수준의 여유율 확보)

□ 네트워크 성능 분석

성능 자동화 도구나 OS 기본 명령어인 netstat 명령어를 이용하여 시스템의 네트워크 Device에 대한 I/O현황을 분석한다.

〈표 4-14〉 네트워크 성능분석

분석 항목	내용	기준치(예시)
네트워크 패킷 비율(%)	- 초당 발생하는 모든 인터페이스에 대한 성공적인 패킷 수(error이나 collision 없이 처리된 inbound/outbound 패킷)	네트워크 사정을 감안한 적정 수준 설정
네트워크 Collision 비율(%)	- 네트워크 인터페이스별 collision 비율	collision 비율 2% 이하
네트워크 트래픽	- 네트워크 인터페이스별 트래픽 양	네트워크 사정을 감안한 적정 수준 설정

나. 성능 개선 방안 수립

(1) 대상 선정

성능 모니터링을 통하여 임계치를 초과하는 현상이 발생할 경우, 또는 성능 데이터 분석을 통하여 사전예방을 목적으로 하는 경우, 성능 개선을 수행하기 위한 개선 대상을 선정한다. 서버 성능 문제는 주로 프로세스 상태, CPU 사용량, 메모리 사용량, 디스크 사용량, 기타 서버 구성요소의 용량, 어플리케이션 및 데이터베이스 설계 등에 따라 결정되며, 이들 자원을 대상으로 성능저하의 원인을 분석하고 그 원인을 제거하기 위한 성능 개선 대상을 선정한다.

성능저하의 원인은 매우 다양하지만 일반적인 원인은 〈표 4-15〉와 같다.

〈표 4-15〉 성능저하의 원인

원인	내용
서버 자원의 부족	충분하지 않은 프로세서 파워, 속도, 부족한 메모리(메모리 누수현상 포함) 느린 Input/Output 부품들이 원인이 된다.
성능조정의 부족	부정확한 파라미터 셋팅, 성능조정을 안 하거나, 잘못된 성능 조정이 원인이 되기도 한다.
I/O 조각화 현상	연속적이지 못하고 조각조각 깨진 형태로 파일이 저장되면, 찾는 작업의 횟수가 기하급수적으로 증가하여서 전체적인 I/O 응답 시간이 증가한다.
데이터 이동	데이터 전송 패킷의 크기 등 전체적인 성능상의 이슈를 제공한다. 이러한 파라미터 값들의 부조화는 종종 심각한 성능저하를 일으킨다.
프로그래밍 오류	설계, 프로그래밍, 구현, 스크립트 등이 원인이 된다. 가장 주요한 원인 중에 하나는 플랫폼에 맞지 않는 잘못된 사용으로 인한 성능저하이다.
데이터베이스 설계의 오류	데이터베이스는 구조적으로 설계되므로 데이터베이스의 재설계 및 구현에 많은 지식과 통찰력, 노력을 요한다. 주요 데이터베이스의 재구축 작업은 비용이 많이 들고 현업업무에 지장을 초래하지만 데이터베이스에 대한 성능 조정을 통해서 비약적인 서버 성능효과를 얻을 수도 있다.

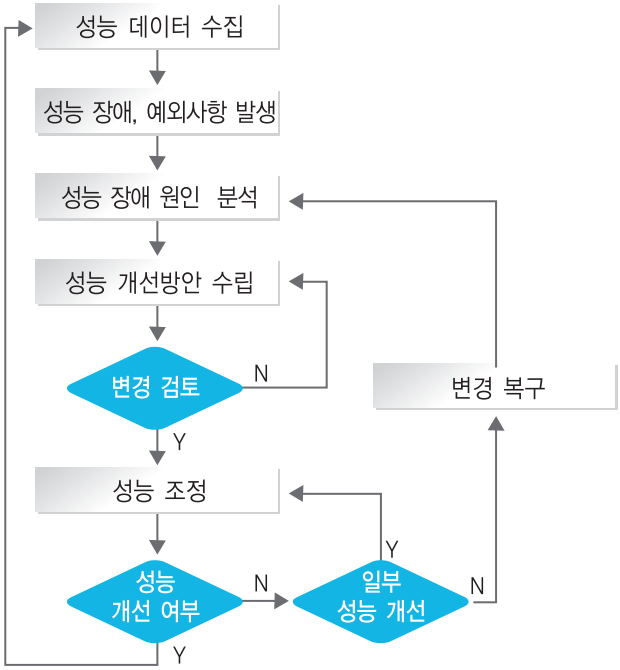
악성코드	바이러스, 웜, 트로이목마, 해킹 툴 등은 의도적이든, 의도적이지 않든 성능에 악영향을 끼친다.
버그	대부분의 어플리케이션, 스크립트, 프로그램 등은 버그(프로그램의 잘못된 사용)를 가지고 있다. 프로그래머, 컴파일러, 코드생성기, 링커(연결프로그램), 인터프리터(해석기), 운영체제는 그들 안에 어느 정도의 에러를 가지고 있다. 이것이 원치 않는 오동작을 일으키기도 한다.
하드웨어 down	하드웨어는 급격히 보다는 서서히 down되는 특징이 있다. 내부 에러 수정 메커니즘에 대해서 요청이 증가하기도 하며, 정품이 아닌 대체용품의 사용으로 인해서 성능저하를 초래할 수도 있다.
외부적 요인	사용자들은 가끔 에러를 발생시켜서 성능을 떨어뜨리곤 한다. 이것은 사용자들에 의해 어플리케이션이 비효율적이고, 잘못 사용된다는 것을 의미한다. 이러한 요인에 대해서는 사용자의 교육과 훈련으로 극복할 수 있다.
구성요소의 부조화	어플리케이션은 구조적으로 한정된 수량의 쓰레드를 구동시킬 수 있다. 어플리케이션이 멀티프로세서 서버로 마이그레이션 될 때, 어플리케이션 아키텍처상 병목현상을 일으킬 수 있다. 예를 들어, 시스템이 40%의 CPU부하를 나타내고, 30%의 메모리 사용하고 있다. 이 경우, 두 개까지만 쓰레드를 동시에 처리하도록 한계가 설정되었다면 더 이상의 부하처리가 안될 수 있다.

(2) 방안 수립

서버 자원들의 성능 기준치를 초과하였을 경우에 해당 자원에 대한 성능 저하의 원인을 파악하기 위하여 의심요소 감시, 문제 발생 요소 파악, 정확한 원인 분석, 타 자원과의 상호작용 분석 등 성능 저하의 원인을 분석하고, 그 원인을 제거하기 위한 성능 개선방안을 도출해야 한다. 도출된 개선방안에 의해서 시스템 구성의 변경사항 발생시 변경관리와 연계하여 변경검토 및 승인 과정을 거쳐 성능 조정을 수행한다.

성능조정 전 충분한 테스트가 이루어져야 하며 성능 조정 수행 후 결과에 대한 검증을 통해 개선 여부를 반드시 확인한다. 개선이 미진한 경우에 대해서는 지속적인 성능 조정을 통하여 개선

을 수행하고, 성능조정이 실패하였을 경우에는 성능 조정 이전 시스템 구성상태로 복구되도록 하여야 한다. 이를 위하여 성능 개선방안에 성능 조정 실패시 이전 상태로 복구를 위한 계획이 포함되어야 한다.



(그림 4-3) 성능 개선을 위한 절차도

(가) 성능 개선 방안 수립

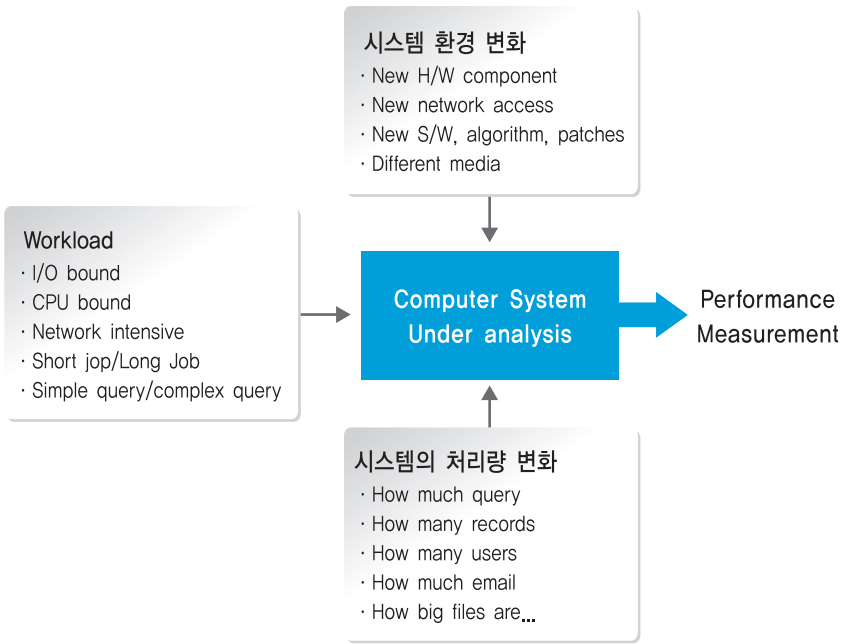
성능 개선을 위해 성능관리자 및 해당 시스템 관련 담당자 즉, 시스템관리자, 어플리케이션 관리자, 데이터베이스 관리자, 네트워크 관리자가 협의하여 개선대상 및 방안을 작성한다. 해당 관련자들의 협의가 이루어지지 않으면 성능 저하의 원인이 정확하게 분석되지 않아 원인 제거가 제대로 될 수 없어 결국 성능 개선을 기대할 수 없다.

또한 개선 효과에 대한 우선순위와 영향도를 고려하여 개선 단계를 선정한다. 일반적으로 어

플리케이션, 데이터베이스, 서버 및 네트워크, OS 파라미터 셋팅의 순으로 성능 조정 수행하여 최종 사용자가 원하는 성능을 제공할 수 있도록 개선방안을 수립하도록 한다.

(나) 성능 영향 요소 분석을 통한 성능 개선 기본 방향 설정

성능 개선을 위해서는 시스템의 성능을 좌우하는 요소들을 카테고리별로 분석할 필요가 있다.



(그림 4-4) 성능 영향요소 분석

(그림 4-4)에서 나타나는 세 가지 카테고리(시스템의 환경변화, 워크로드의 특성, 시스템의 처리량 변화)가 전반적인 시스템의 성능을 좌우하며, 따라서 이러한 각 요소들에 대해서 정확한 파악을 통하여 성능 개선 기본 방향을 설정한다.

(다) 성능 개선방안 수립을 위해 파악해야 할 기초자료

- 시스템 구성 환경(H/W 구성도, CPU 타입, 디스크 타입 등)
- 데이터베이스 환경 구성 및 사용현황
- 주요 어플리케이션 목록 및 사용현황
- 네트워크 구성 및 사용현황
- 시스템 구성 변경사항
- 워크로드의 특성
- 시스템의 처리량 변화
- 시스템 사용자 수 및 동시 사용자 수
- 프로세스 종류 및 사용 상태
- 성능 장애 및 예외 사항 보고서

다. 성능 개선 실행 및 검증

(1) 성능 조정

서버의 구성요소는 다양한 성능 특성, 특징을 가지고 있으며, 이에 따라 운영 환경에서 다양하게 동작한다. 서버자원의 성능 조정은 이러한 다양성을 인식하고 고려해야 한다. 또한 성능관리 프로세스는 서비스수준관리, 용량관리, 구성 및 변경관리 등 타 프로세스와의 연계성을 고려하여 성능관리 운영상의 어떠한 조치도 타 프로세스와 협업해서 처리해야 한다.

앞서 성능 저하를 일으키는 원인들에 대하여 설명하였고, 다음은 성능 조정을 위하여 몇 가지 유의해야 할 사항에 대해서 예를 들어 설명한다.

- 다양한 구성요소가 포화상태(한계치)에 도달하면, 전혀 다른 방식으로 동작한다. 예를 들면, 프로세서가 50% 정도의 성능밖에 나오지 않는 경우 이것은 어플리케이션 프로그램 및 운영체제 명령어가 50% 정도 밖에 처리되지 않는다는 것을 의미한다. 그리고, 나머지 50%는 동작하지 않는 상태에 이르게 된다. 다시 말해서 시스템 모니터링 도구에서 프로세서가 100%를 사용한다고 가리키고 있는 것은 현재 추가로 100%의 프로세스 성능이 더 필요하다는 것을 의미한다. 이 경우에 프로세서가 처리하기 위한 프로그램 코드가 큐(queue)에 대기하고 있다는 것이며, 프로세싱 파워가 부족하다는 것을 뜻한다.

- 특정한 상황에 대해서 적절한 측정 작업이 이루어져야 한다. 프로세서의 부하가 90% 정도라면 걱정하지 않아도 된다. 왜냐하면 프로세서 부하가 5% 정도와 비교하면 단위명령코드의 실행속도는 같기 때문이다(그리고, 큐에서 미처리 되고 쌓이는 현상도 없다).

- 대부분의 프로세서 부하가 100%를 일으키는 원인은 잘못된 프로그래밍 때문이다. 원인을 제공한 프로그램은 아마도 무한루프에 빠져 있을 가능성이 크다. 어떤 특정 조건이 되면, 작은 어플리케이션 코드가 계속 실행이 된다. 어플리케이션 코드에서 이러한 예외사항에 대한 처리를 해준다면 다시 문제가 발생하지 않지만, 루프를 돌면서 계속 코드가 실행되면 시스템 모니터링 툴은 100%의 프로세서 부하를 나타낼 것이다. 이는 프로세서 공유 메커니즘으로 인해서, 현재 같은 프로세서에서 실행되고 있는 다른 프로그램들에 악영향을 끼칠 수 있다. 이로 인해서 시스템 응답속도가 저하되고, 시스템이 동작을 멈추게 된다. 성능관리자는 루프를 도는 프로세스를 강제로 종료하는 (Killing) 조치를 수행하고, 메모리 영역의 내용을 프린트로 출력해서 문제가 되는 코드를 분석하는 조치를 수행하게 된다.

- 특정 구성요소에 의한 시스템 성능 저하메모리가 포화상태에 다다르면 시스템의 응답속도는 느려진다. 느려지는 원인은 대부분의 메모리가 가상메모리의 형태로 운영되기 때문이다. 가상메모리는 종종 느리고 저렴한 구성요소인 하드디스크에 내부 메모리가 복사된다. 이는 단순히 복사일 뿐이지만 메모리에 대한 다수의 요청이 발생 할수록, 가상메모리의 장애가 발생하게 된다. 이러한 장애는 메모리 (실제메모리)와 하드디스크 (가상메모리)의 처리속도 차이가 발생함으로 인해서 일어날 수 있다.

성능관리자는 성능저하의 원인이 시스템의 다양한 연관관계 속에서 발생하므로 이러한 점들을 고려하여 성능 조정을 위한 방법들을 강구하여야만 한다.

(가) 응답속도의 문제점을 낮추기 위한 여러 가지 방법들

▶ 구성 조정하기

- 구성 조정 및 가상메모리 SETUP 조정하기

▶ 캐싱

- 자주 사용되는 메모리 데이터 영역의 확보
- 프로세서와 메모리간의 데이터 전송에 쓰이는 캐시 영역도 있다.

▶ 작업로드의 균형 조정

- 트랜잭션은 시작된 위치에 따라 특정 게이트웨이에 있는 서버에 도달할 수 있다. 게이트웨이에 대한 시작 지점 비율을 균형적으로 재분배함으로써 성능을 조정할 수 있다.

▶ 효율적인 메모리 사용

- 상황에 따라 메모리를 더 많이 또는 더 적게 사용하도록 한다. 데이터가 메모리에 읽혀져서 조작되면 한 프로세스가 파일을 통해 순차적으로 읽어 올 때보다 더 효율적으로 리소스를 사용할 수 있다. 또한 많은 프로세스가 메모리 리소스를 사용하기 위해 경합할 수도 있다. 과도한 요구는 CPU 사용률 증가 및 메모리에 대한 페이지 스왑(Page Swap) 지연을 초래한다.

▶ 스트라이핑 하드디스크

- 읽기와 쓰기가 동시에 일어나는 하드디스크의 영역을 논리적으로 구분해서 나누어 사용함으로써 가상메모리의 속도를 향상시킨다.
- 스트라이핑은 RAID에서도 사용되는 기술이다.

▶ 압축

- 데이터 전송 시 데이터를 압축해서 전송하면 압축해제 하는데 시간이 더 많이 걸린다. CPU 용량이 여유가 있을 경우 사용한다.

(나) 점점 감소하는 성능 조정 효과의 법칙

최대성능의 80%로 운영되는 구성요소가 있다고 가정한다면 첫 번째 튜닝은 사용하지 않고 있는 잠재적인 성능(20%)의 반 정도를 얻어낸다. 또다시, 두 번째 튜닝은 첫 번째 튜닝결과에서 다시 사용하지 않고 있는 잠재적인 성능(10%)의 반 정도를 얻어낸다. 세 번째 튜닝은 두 번째 튜닝결과에서 다시 사용하지 않고 있는 잠재적인 성능(5%)의 반 정도를 얻어낸다. 즉, 튜닝효과는 점점 줄어든다.

예를 들면, 매번 튜닝하는 노력과 비용이 같다고 가정한다.

- 첫 번째 튜닝은 10%의 이득을 얻었다(사용하지 않고 있는 20%의 절반)
- 두 번째 튜닝은 5%의 이득을 얻었다(사용하지 않고 있는 10%의 절반)
- 세 번째 튜닝은 2.5%의 이득을 얻었다(사용하지 않고 있는 나머지 5%의 절반)

경제적인 효과는,

- 두 번째 튜닝은 첫 번째 튜닝에 비해서 2배로 비싸다.
- 세 번째 튜닝은 첫 번째 튜닝에 비해서 4배로 비싸다.

간단히 정리하면, 구성튜닝이 계속될수록 얻는 효과는 점점 감소하게 된다.

(다) OS 파라미터 셋에 의한 성능조정

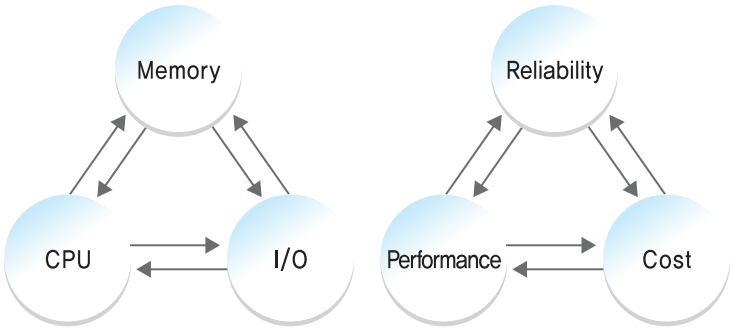
성능조정은 자원 활용의 과정에서 자원 최적화로부터 얻어지는 액티비티이며, 이것으로부터 성능개선이 이루어진다. 파라미터값의 셋팅으로, 서버의 성능향상을 가능하게 할 수 있다. 이러한 꾸준한 성능조정을 통해서 성능 품질이 향상될 수 있다. 만약 특별한 요구사항이나 요청이 있다면, 서버의 OS의 파라미터셋 적용이 필요하다. 파라미터셋이 변경되기 전에 파라미터셋이 기능성에 어떠한 영향을 끼치는지, 그리고 요구되는 기능성, 작업들에 대해서 명확히 이해하는 것은 아주 중요한 사안이다. 같은 타입의 서버라고 할지라도 요구되어지는 기능에 따라서 파라미터값은

완전히 다르게 설정되어 진다. 서버 각각의 구성요소들은 일반적 목적을 가지고 있다. 공급자나 제조자는 이 구성요소가 어떠한 위치에서 어떠한 방식으로 운영되는지를 정확히 알지는 못한다. 각 구성요소가 어떠한 기능을 수행하는지 정확히 아는 것은 서버 운영자와 성능관리자의 책임이다. 그러므로, 각각의 구성요소는 요구되어지는 기능을 효과적으로 수행하도록 성능조정이 되어야 한다. 초기에는 이러한 요구사항을 만족할 수 있도록 서버 파라미터 셋팅이 이루어진다. 이것은 일정 시간 동안 만족스러운 결과로 나타나겠지만, 특정 서버가 시간을 거치면서 생기는 성능저하, 구성요소의 교체, 서버 활용방법이나 서비스의 변경 등에 의해 성능조정을 지속적으로 수행하여야 한다.

성능조정이나 성능관리 도구를 제공하는 업체는 많이 있다. 이러한 도구들은 특정 플랫폼에 영향을 받으며, 플랫폼 제조업체로부터 제공되므로 각 기관들은 시스템 환경에 적합한 도구를 선정하여 활용하도록 한다.

(라) 인프라스트럭쳐 병목현상

일부 서버 운영자나 성능관리자들은 한두 가지 관리대상 항목의 성능을 최고로 끌어올리는데 많은 관심과 노력을 기울인다. 하지만 시간과 자원의 비경제적인 성능조정으로 인하여 비생산성이 나타날 수 있다. 한군데에서 병목현상을 완화해도 순차적으로 다음구역으로 병목현상이 전이될 수 있기 때문이다. 다시 말해서 복잡한 서버 구성은 연속적으로 병목현상이 있다는 것을 고려해야만 한다. 이러한 결과로 인하여 병목현상은 서버의 정상적인 운영을 방해한다. 이것은 또한 충분하지 못한 대역폭과 속도에 기인한다. 비록 충분한 대역폭이 있다고 하더라도 시작단부터 끝단까지의 속도가 여전히 느릴 수 있다. 또한, 속도를 두 배로 늘린다 하더라도, 어떠한 경로에 병목현상으로 인해서 충분한 성능이 나오지 않을 수도 있다. 그러므로 성능관리자는 서버 성능조정을 위해서 전체 인프라스트럭쳐의 서로 관련된 여러 가지 구성요소에 대하여 고려하여 연관된 구성요소에 대한 성능 조정을 반복적으로 수행하여 성능 목표의 달성여부를 확인하여야 한다.



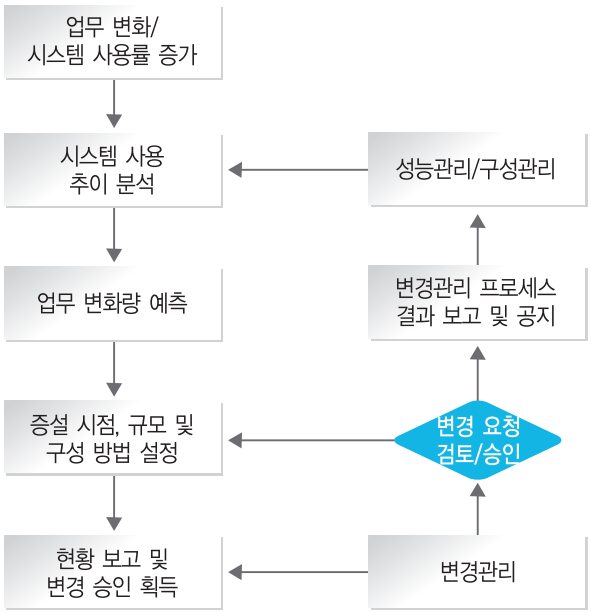
(그림 4-5) 구성요소들의 연관관계를 고려한 성능 조정

〈표 4-16〉 구성요소별 성능 조정 방법(예시)

구성요소	조치내용
CPU	- 사용자 프로세스의 상태 분석 - 불필요한 프로세스 종료(KILL) - 처리가 늦어도 되는 프로세스 PRIORITY조정 - JOB 처리시간 조정
메모리	- 메모리 이용상태 분석 - 메모리를 많이 사용하는 프로세스 확인 후 조치 - BUFFER CACHE 사용률 분석 후 상호 조정 - 지속적인 SWAPPING 발생시 메모리 확장
디스크	- 디스크 I/O상태 분석 - 디스크 간 I/O 부하 분산여부 파악 - 메모리 여유율 파악 후 BUFFER CACHE 증가 - 디스크 자원 재배치를 통한 부하 분산('/' 가 포함된 디스크의 병목현상은 전체 시스템에 영향을 줌) - 특정 I/O CHANNEL에 부하 집중 시 I/O CHANNEL 증설 후 분산 실시
BUFFER CACHE	- 메모리 사용률을 고려하여 BUFFER SIZE 증가
기타	- 성능향상을 위한 PATCH 존재 시 적용 - 시스템간 서비스 분산 고려 - 디스크 I/O 단위를 고려하여 어플리케이션의 I/O 단위를 설정 - 시스템 커널은 연관된 파라미터를 고려하여 수정

(2) 용량 증설

성능관리자는 성능조정으로 개선효과가 부족하다면 용량증설을 고려할 수 있다. 사용자수, 응용프로그램이나 서비스의 사용자가 증가하면 서버 성능 저하 현상이 나타나므로 해당 시스템의 용량을 증설해야 한다. 서버 용량 및 하드웨어와 소프트웨어의 구성, 전산 인프라를 업그레이드 하는 방법으로 서비스 용량을 늘려, 더 많은 사용자나 더 복잡한 응용프로그램 솔루션을 허용할 수 있다. 하지만 이러한 용량증설이 부적합한 인프라 구성의 결과나 절차에 의하지 않는 변경 때문일 수도 있다는 것을 염두에 두고 신중히 검토되어야 한다.



(그림 4-6) 용량증설 절차

(가) 용량 사용 예측

용량 사용 예측은 일시적인 시스템 자원 사용현황을 조사하는 것이 아니라 정기적이고 장기간 동안 모니터링 되고 수집된 시스템의 성능 데이터를 분석해야 한다. 그 분석된 결과를 토

대로 미래의 작업로드를 예측하고 미래에 필요한 용량을 산정하여 서전에 용량에 대한 대비를 함으로써 부족한 자원으로 인한 서버 성능 저하를 줄여 효율적이고 안정적인 시스템 환경을 유지하도록 한다.

(나) 용량 산정

정보시스템의 서버, 디스크, 네트워크의 용량을 산정하기 위하여 기존 업무와 사용자 접속 환경의 특성을 바탕으로 동시사용자, 트랜잭션 발생량 등을 고려하여 최적의 용량을 산정한다. 산정기준은 OLTP 벤치마크인 TPC-C(tpmC)를 적용한다. 보다 자세한 서버 용량산정 기준은 '정보시스템 규모별 용량산정 기준 연구(2004, 한국전산원)' 를 참조한다(문서번호 : NCA IV-RER- 04019/2004.10).

(다) 용량 산정시 고려사항

▶ 서버

- 서버 H/W의 증설 가능 용량 / 동시사용자 및 트랜잭션 양을 고려
- 향후 업무 및 사용자 증가, 연계기관 확대를 고려 - 여유율 20~30% 적용
- 총 요구 tpmC 대비 제작사의 예상 tpmC 자료와 비교 후 기종 및 CPU 산정시스템 운영 소프트웨어(백업, 보안, SMS, 효율분석)
- 서버 용량 증설에 필요한 전산기계실 상면, 전원, 하중 등

▶ 디스크

- DBMS, 어플리케이션의 디스크 요구량 고려
- 여유율 20~30% 적용
- 업무 증가분, 연계기관 확대에 따른 데이터 증가치 고려
- 가능한 디스크 구성방식 확인 / 파일시스템 구성 시 여유율 적용

4 2.2 네트워크

가. 데이터 수집 및 분석

(1) 측정항목

네트워크 관리자는 최적의 네트워크 성능 유지 및 네트워크 장애 발생을 사전에 방지하기 위하여 정기적인 점검을 실시해야 한다. 점검 내용은 〈표 4-17〉을 기준으로 하며 모든 점검내용은 기록관리 되어야 한다.

〈표 4-17〉 성능관리 항목별 관점의 측정 항목

측정항목	측정 요소	측정 방법	측정주기
가용성(Availability)	네트워크 장비	PING	실시간
응답시간(Response Time)	네트워크 장비	PING 응답시간	실시간
	어플리케이션	패킷 캡쳐	실시간
장애율(Error Rate)	회선(LAN-collision)	SNMP	실시간
	회선(WAN)	SNMP	실시간
어플리케이션 분포	어플리케이션	패킷 캡쳐	실시간
사용률(Utilization)	회선(LAN/WAN)	SNMP/패킷 캡쳐	실시간
	네트워크 장비 (CPU, Memory, Buffer)	SNMP	실시간

다음으로 기술할 네트워크 자원별 관점의 측정 항목 중 각각의 표에서 제시한 각종 임계치는 각 기관의 비즈니스 환경이나 어플리케이션 및 네트워크 장비의 특성에 따라 그 임계치의 설정을 조절해야 한다. 따라서 아래에서 제시하는 수치는 일반적인 환경의 임계치이며, 단지 참조사항으로 활용될 수 있다.

(가) WAN 성능

WAN 링크와 관련된 데이터로는 회로 사용률, 오류, 포기 및 브로드캐스트/멀티캐스트가 있다. 〈표 4-18〉에서는 모니터 할 일반적인 임계치 및 이벤트를 보여준다.

〈표 4-18〉 WAN 임계치 및 이벤트

	평균사용률(%)	최고사용률(%)	오류(%)	포기(%)	브로드캐스트 멀티케스트
경고 임계치	〉30	〉90	〉0.1	〉0.1	200
치명적 임계치	〉70	〉98	〉1	〉1	300

▶ 사용률 : 평균 사용률은 WAN 링크가 최대 용량에 가까워지고 있음을 나타내는 WAN 사용률 데이터를 이용한다. 각 폴링 간격에서, 보고서에는 입력 사용률이나 출력 사용률 중에서 큰 것을 표시한다. 일부 WAN 링크는 주로 한 방향으로만 사용된다. 한 방향으로만 너무 많이 사용하면 사용자 성능 문제와 업그레이드 요구 사항에 영향을 준다.

최고 사용률은 또한 회로에서 소통량이 최고조에 도달하는 시점인 가장 큰 수집 간격 피크도 보여준다. 평균 사용률은 낮지만 최고 소통량이 자주 발생하는 네트워크의 경우 솔루션을 구성하거나 구축해야 한다(예: 프레임 릴레이 또는 ATM). 전반적인 사용률 통계(송수신 모두)가 작으면 해당 회로에서의 대역폭을 줄여 비용을 절약할 수 있다. 사용률 통계가 크면 이는 해당 세그먼트에서 대역폭을 늘려야 하는 구체적인 근거가 된다.

▶ 오류율 : WAN 오류율은 주로 회선 상태와 관련된다. 이 통계는 기본적으로 직렬 데이터 링크가 양호한지 모니터하기 위해 사용된다. 이 값은 보고 오류가 증가하고 있음을 파악하여 미리 수정할 수 있다. 즉, 고치기 힘든 장애가 될 때까지 기다리지 않는다. 따라서 채널 서비스 장치/디지털 서비스 장치(CSU/DSU), 케이블, 임대 회선 및 인터페이스 카드가 완전히 못쓰게 되기 전에 문제를 식별할 수 있다.

▶ 패킷 포기 : 이 값은 대역폭 사용량이 과도하다는 것을 조기에 알려줄 수 있다. 프레임보다 빨리 채워지는 버퍼가 WAN 링크로 전달되거나 WAN 링크에서 전달되면 라우터는 프레임을 포기하기 시작한다. 따라서 링크 용량이 라우터의 CPU 용량을 초과할 상황이 되거나 이미 초과하고 있으면 포기를 나타내는 라우터 인터페이스의 사용률을 점검해야 한다. 간헐적으로 일어나는 몇몇 포기 이벤트가 네트워크 상태가 양호하지 않음을 의미하지는 않는다.

그러나 이러한 이벤트는 링크가 포화 상태에 도달하고 있다는 신호일 수 있다. 지속적으로 포기가 일어나거나 이벤트에 수 천 개의 포기가 포함되어 있는 경우 중요한 문제가 있음을 뜻한다. 따라서 시간이 경과함에 따라 포기를 추적하여 네트워크의 상태를 평가하고 라우터나 링크에 과도한 로드가 걸리고 있음을 조기에 경고해주는 것이 중요하다.

▶ 브로드캐스트/멀티캐스트 : 이 값은 초당 브로드캐스트/멀티캐스트 수를 나타낸다. 값이 증가하거나 갑자기 최고치에 도달하면 동적 라우팅 네트워크의 라우팅 프로토콜 오버헤드 문제가 있음을 의미할 수 있다.

(나) 프레임 릴레이 분석

프레임 릴레이 영구 가상 회로(PVC)의 상태 데이터로는 회로 사용률, 혼잡 및 포기 타당성 등이 있다. 〈표 4-19〉에서는 모니터 할 일반적인 임계치 및 이벤트에 대한 예를 보여준다.

〈표 4-19〉 프레임 릴레이 임계치 및 이벤트

	평균사용률(%)	최고사용률(%)	최고 BECN	최고 FECN	오류
경고 임계치	〉80	〉150	〉100	〉100	3000
치명적 임계치	〉100	〉200	〉500	〉500	5000

▶ 사용률 : 평균 사용률은 가상 회로가 최대 용량에 가까워지고 있음을 나타내는 사용률 데이터를 이용한다. 각 폴링 간격에서, 보고서에는 입력 사용률이나 출력 사용률 중에서 큰 것을 표시한다. 일부 PVC는 주로 한 방향으로만 사용된다. 한 방향으로만 너무 많이 사용하면 사용자 성능 문제와 업그레이드 요구 사항에 영향을 준다. 최고 사용률은 또한 회로에서 소통량이 최고조에 도달하는 시점인 가장 큰 수집 간격 피크도 보여준다. 일부 회로에서는 높은 사용률을 처리할 수 있는 구성이 필요할 수도 있다.

▶ 최고 BECN/FECN : 프레임 릴레이 환경에서의 혼잡은 순방향 명시적 혼잡 알림(FECN) 및 역방향 명시적 혼잡 알림(BECN) 형태로 나타난다. 이러한 알림은 한 방향에서 다른 방향으로의 데이터 흐름 속도가 늦어지고 있다는 것을 알려주는 인터네트워킹 장치의 역할을 한다. 이 통계는 패킷으로부터 데이터가 수집되는 라우터의 경로에 따라 혼잡이 있으며, 더 나쁘게는 패킷에서 이 혼잡 비트가 포함된 다른 패킷이 수신되지 않는 것으로 표시되었음을 라우터에게 알려준다. 다양한 PVC의 양 끝을 검사하여 패킷 손실이 발생했는지 확인하고 네트워크 디자인을 검토하거나 네트워크에서 혼잡이 발생하는 원인에 대해 프레임 릴레이 공급자에게 문의할 수도 있다. 이러한 링크에서의 통신에 대해서는 BECN 및 FECN 값을 자세히 모니터하여 품질을 유지하기 위한 조정이 필요한지 확인해야 한다.

▶ 오류 : 오류 메트릭은 백만 개의 프레임 당 오류 수를 측정한다. 단순 네트워크 관리 프로토콜(SNMP) 에이전트에 대한 대부분의 프레임 에이전트는 오류 통계를 제공하지 않는다. 오류율은 주로 회선 상태와 관련된다. 이 통계는 기본적으로 링크가 양호한지 모니터하기 위해 사용된다. 이 값은 오류가 증가하고 있음을 파악하여 미리 수정할 수 있다. 즉, 고치기 힘든 장애가 될 때까지 기다리지 않는다. 따라서 채널 서비스 장치/디지털 서비스 장치(CSU/DSU), 케이블, 임대 회선 및 인터페이스 카드가 완전히 못쓰게 되기 전에 문제를 식별할 수 있다. 음성을 전달하는 링크인 경우 손실로 인해 품질이 저하될 수 있으므로 가능하면 이들 통계를 자세히 모니터해야 한다.

(다) 프레임 릴레이(CSU/DSU) PVC 분석

프레임 릴레이 서비스 인식 CSU/DSU와 관련된 데이터이다. 이러한 장치에는 자세한 네트워크 관리 통계가 포함되어 있다. 회선 사용률, 혼잡, 포기 타당성 및 삭제된 패킷 등과 같은 프레임 릴레이 PVC와 관련된 종단 간 통계를 이러한 장치로부터 수집할 수 있다. 〈표4-20〉에서는 모니터 할 일반적인 임계치와 비율 및 패킷 수에 대한 예를 보여준다.

〈표 4-20〉 프레임 릴레이 CSU/DSU 임계치, 비율 및 패킷

	평균사용률(%)	최고사용률(%)	삭제된 패킷(#)	평균 RTD	최고 RTD
경고 임계치	〉80	〉150	〉25	〉100	〉200
치명적 임계치	〉100	〉200	〉100	〉200	〉400

- 사용률 : 평균 사용률은 가상 회로가 최대 용량에 가까워지고 있음을 나타내는 사용률 데이터를 이용한다. 각 폴링 간격에서, 보고서에는 입력 사용률이나 출력 사용률 중에서 큰 것을 표시한다. 일부 PVC는 주로 한 방향으로만 사용된다. 한 방향으로만 너무 많이 사용하면 사용자 성능 문제와 업그레이드 요구 사항에 영향을 준다. 최고 사용률은 또한 회로에서 소통량이 최고조에 도달하는 시점인 가장 큰 수집 간격 피크도 보여준다.

- 삭제된 패킷 : 서비스 인식 CSU/DSU들 사이에 지능과 통신이 추가되어 두 방향 중에 포기된 패킷이 있는지 여부를 일정 기간 이상 동안 인식할 수 있다.

- 왕복 지연(RTD) : 프레임 서비스 인식 CSU/DSU에는 각 장치 사이(예: 사이트 사이)의 네트워크 대기 시간을 모니터할 수 있는 기능이 있다. 이를 통해 프레임 릴레이 내의 네트워크 지연을 인식할 수 있다.

(라) ATM 트렁크 분석

ATM 트렁크 회로와 관련된 데이터에는 회로 사용률 및 셀 손실률이 포함된다. 〈표 4-21〉에서는 모니터 할 일반적인 임계 비율의 예를 보여준다.

〈표 4-21〉 ATM 임계치

	평균사용률(%)	최고사용률(%)	셀 포기율(%)
경고 임계치	〉40	〉95	〉0.1
치명적 임계치	〉50	〉98	〉1

▶ 사용률 : 평균 사용률은 WAN 링크가 최대 용량에 가까워지고 있음을 나타내는 WAN 사용률 데이터를 이용한다. 각 폴링 간격에서, 보고서에는 입력 사용률이나 출력 사용률 중에서 큰 것을 표시한다. 일부 WAN 링크는 주로 한 방향으로만 사용된다. 한 방향으로만 너무 많이 사용하면 사용자 성능 문제와 업그레이드 요구 사항에 영향을 줄 수 있다.
최고 사용률은 또한 회로에서 소통량이 최고조에 도달하는 시점인 가장 큰 수집 간격 피크도 보여준다. 이 정보를 사용하여 최적화를 위한 적절한 회로를 구성할 수도 있다.

▶ 셀 포기율 : 이 값은 포기한 셀을 전체 셀 수에 대한 비율로 측정한다. 리소스 부족으로 인해 포트 수준에서 셀이 손실될 수 있다.

(마) ATM VC 분석

ATM PVC와 관련된 데이터에는 회로 사용률 및 셀 손실율이 포함된다. 〈표 4-22〉에서는 모니터 할 비율의 예를 보여준다.

〈표 4-22〉 ATM PVC 비율

	평균사용률(%)	최고사용률(%)	셀 오류율	셀 거부율
경고 임계치	〉50	〉95	〉0.1	〉0.1
치명적 임계치	〉60	〉98	〉1	〉0.3

- 사용률 : 평균 사용률은 가상 회로가 최대 용량에 가까워지고 있음을 나타내는 PVC 사용률 데이터를 이용한다. 각 폴링 간격에서, 보고서에는 입력 사용률이나 출력 사용률 중에서 큰 것을 표시한다. 일부 가상 회로는 주로 한 방향으로만 사용된다. 한 방향으로만 너무 많이 사용하면 사용자 성능 문제와 업그레이드 요구 사항에 영향을 준다.

  최고 사용률은 또한 회로에서 소통량이 최고조에 도달하는 시점인 가장 큰 수집 간격 피크도 보여준다.

- 셀 오류율 : 이 값은 측정한 오류 셀을 전체 셀 수에 대한 비율로 측정한다. 일반적으로 이 값은 셀에 헤더 오류 검사값(HEC) 오류나 잘못된 VPI 및 VCI가 있다는 것을 나타낸다.

- 셀 거부율 : 이 값은 거부된 셀을 전체 셀 수에 대한 비율로 측정한다. 사용량 매개 변수 제어, 조기 패킷 포기, 임의적인 패킷 포기 등과 같은 소통량 관리 때문에 셀이 거부(포기)될 수 있다.

(바) LAN 분석

LAN 세그먼트와 관련된 데이터에는 대부분의 라우터, 스위치, 허브 및 서버에서 제공되는 일반적인 통계 데이터가 포함된다. 이 데이터를 보면 링 사용률, 오류 및 포기에 대해 자세히 알 수 있다. 〈표 4-23〉에서는 모니터 할 비율의 예를 보여준다.

〈표 4-23〉 LAN 분석 비율

	평균사용률(%)	최고사용률(%)	오류(%)	포기(%)	브로드캐스트/멀티케스트
경고 임계치	〉10	〉25	〉0.1	〉0.1	200
치명적 임계치	〉70	〉50	〉1	〉1	300

▶ 사용률 : LAN 사용률은 MIB를 통해 수집되며 다양한 세그먼트 인터페이스를 통과하는 LAN 소통량을 보여준다. 평균 사용률 데이터는 설정된 주요 시간대에 발생한 네트워크 소통량을 알려준다. 최고 사용률 데이터는 사용률을 높이는 과다한 작업이 있음을 나타낸다. 이 정보는 LAN에서 세그먼트 크기를 조정하고 세그먼트를 다양하게 변경할 경우 유용하다.

▶ 오류율 : 평균 오류율은 시간당 평균 오류율을 보여준다. LAN 오류율은 기본적으로 미디어 유형, 설치 품질 및 로드와 관련된다. 10Base-T 등의 비차폐 꼬아진 쌍(UTP) LAN에서는 고품질 와이어링, 구성 요소 및 설치 기술을 사용하지 않으면 오류가 발생할 가능성이 크다. 오류가 발생하면 소통량이 증가함에 따라 비율도 증가할 가능성이 있다.

이 데이터는 RMON 프로브가 없는 LAN 상태를 모니터할 때 특히 유용하다. 라우터에는 라우터로 주소가 지정된 모든 패킷과 관련된 오류가 포착된다. 이는 LAN 세그먼트의 전체 오류율과 비슷하므로, 오류가 있는 케이블, 트랜시버 및 인터페이스 카드에 문제가 발생하기 전에 문제를 발견하는 데 도움이 된다.

▶ 패킷 포기 : 이 값은 대역폭이 초과 사용되고 있음을 조기에 알려준다. 프레임보다 빨리 채워지는 버퍼가 LAN 링크로 전달되거나 LAN 링크에서 전달되면 라우터는 프레임을 포기하기 시작한다. 예를 들어, Cisco 라우터에서의 일반적인 포기 수는 Cisco의 버퍼 없음 카운터와 출력 삭제 카운터의 합계이다.

버퍼 없음 카운터는 주 시스템에 버퍼 공간이 없어서 포기한 수신 패킷 수를 나타낸다. 이 값은 LAN에서의 과다한 브로드캐스트 및 직렬 회선에서의 많은 노이즈로 인해 나타나는데, 패킷 수가 너무 많으면 라우터 작동에 문제가 생기게 된다.

출력 삭제 카운터는 대상 인터페이스의 과도한 사용으로 인해 포기된 출력 패킷 수를 나타낸다. 이 경우 문제는 링크의 과도한 로드이므로, 정기적인 라우팅 업데이트를 줄이도록 라우터를 구성하거나 과도하게 사용되는 프로토콜의 빠른 전환 기능을 해제하거나 궁극적으로는 링크의 대역폭을 늘리는 방법으로 이 문제를 해결해야 한다.

간헐적으로 일어나는 몇몇 포기 이벤트가 네트워크 상태가 양호하지 않음을 의미하지는 않는다. 그러나 이러한 이벤트는 링크가 포화 상태에 도달하고 있다는 신호일 수 있다. 지속적으로 포기가 일어나거나 이벤트에 수천 개의 포기가 포함되어 있는 경우 중요한 문제가 있음을 뜻한다. 따라서 시간이 경과함에 따라 포기를 추적하여 네트워크의 상태를 평가하고 라우터나 링크에 과도한 로드가 걸리고 있음을 조기에 경고해 주는 것이 중요하다.

※ 포기가 시작될 때쯤이면 이미 성능이 저하된 상태에 있다. 성능에 영향을 주기 전에 과도하게 사용된 대역폭을 찾아내려면 LAN 세그먼트에 대한 평균 사용률 및 최고 사용률을 모니터해야 한다.

▶ 브로드캐스트/멀티캐스트 : 이 값은 초당 브로드캐스트/멀티캐스트 수를 나타낸다. 높은 비율의 브로드캐스트 소통량이 반드시 문제가 있음을 나타내는 것은 아니다. 값이 늘어나거나 갑작스럽게 피크에 도달하는 것은 LAN 하드웨어, 응용 프로그램 또는 계층 2 라우팅 문제(예: IPX SAP 폭풍) 등의 문제를 나타낼 수 있다. 이는 또한 대역폭을 불필요하게 사용하고 있다는 신호이기도 하다. 이 숫자는 링크 크기에 따라 다르다.

(사) 이더넷/RMON 분석

이더넷 세그먼트의 상태 관련 데이터에는 이더넷 RMON 프로브에서 제공하는 또 다른 매우 구체적인 통계 데이터가 포함된다. 이 데이터를 보면 링 사용률, 오류 및 이벤트에 대해 자세히 알 수 있다. 또한 RMON은 어떤 종류의 오류가 링에서 일어나고 있는지 알려주는 기능도 가진다. 〈표 4-24〉에서는 모니터 할 비율의 예를 보여준다.

〈표 4-24〉 이더넷/RMON 비율

	평균사용률(%)	최고사용률(%)	오류(%)	충돌률(%)
경고 임계치	〉15	〉25	〉0.1	〉5
치명적 임계치	〉20	〉40	〉1	〉10

Fast Ethernet에서도 동일한 요소가 모니터링 되지만, 이 경우에는 임계치가 조정된다.

〈표 4-25〉 Fast Ethernet 비율

	평균사용률(%)	최고사용률(%)	오류(%)	충돌률(%)
경고 임계치	〉25	〉40	〉0.1	〉5
치명적 임계치	〉50	〉80	〉1	〉10

▶ 사용률 : LAN 사용률은 구식 LAN 이더넷 허브 장치에서 일반적으로 사용이 가능하던 레거시 SNMP 에이전트이다. 사용률 패턴을 관찰하면 허용되는 수준 이내로 소통량 로드가 유지되도록 하고 해당 수준의 추세를 관찰할 수 있다. 이러한 추세를 통해 잠재적인 문제가 사용자에게 영향을 미치기 몇 달 전에 그러한 문제를 파악할 수 있다.

평균 사용률 통계는 네트워크의 소통량이 어느 정도인지에 대한 정확한 시각을 제공한다.

그러나 짧은 기간 동안 활동이 폭증하는 현상은 이 통계로는 알 수 없다. 그러한 폭증을 보여주는 것은 최고 사용률 통계이다. 최고 사용률 통계는 라우터와 직렬 회선의 크기를 조정하고 로컬 구성을 변경할 때 유용한 정보이다.

- 오류율 : LAN 오류율은 대개 미디어 유형, 설치 품질 및 로드와 관련된다. 이더넷 LAN에서 충돌로 인해 오류가 발생할 수 있다. 이더넷에 사용되는 CSMA/CD 액세스 방법에서 충돌은 정상적인 현상이다. 오류가 발생하면 소통량이 증가함에 따라 비율도 증가할 가능성이 있다. 또한 오류가 높은 사용률과 어떤 상관 관계를 보이는지 확인하기 위해 평균 및 최고 사용률 통계를 고려해야 한다. 그렇지 않다면 장애가 굳어지거나 네트워크가 오작동하기 전에 복원해야 하는 케이블 배선 오작동이나 하드웨어 오류가 있는 것일 수 있다.

  Jabbers, undersized, oversized, CRC 등과 같은 발생한 오류 유형을 파악하려면 오류를 추가로 분석해야 한다. 또한 오류가 높은 사용률과 어떤 상관 관계를 보이는지 확인하기 위해 평균 및 최고 사용률 통계를 고려해야 한다. 그렇지 않다면 장애가 굳어지거나 네트워크가 오작동하기 전에 복원해야 하는 케이블 배선 오작동이나 하드웨어 오류가 있는 것일 수 있다.

- 충돌률 : 이더넷 세그먼트에서 시간 프레임당 충돌 수이다. 이더넷에서 충돌은 정상적인 현상이다. 이더넷은 노드에서 패킷이 충돌할 경우 효율적으로 다시 전송하도록 설계되었다. 그러나 소통량이 많아지면 한 노드가 통신 중일 때 다른 노드가 전송을 시도하려는 경우가 더 많이 발생한다. 충돌 수는 네트워크 소통량이 최대가 될 때까지 증가할 수 있다. 응답이 감소하는 시점은 네트워크 크기, 하드웨어 및 소통량에 따라 다르다.

  케이블 연결 길이가 짧고, 리피터와 노드 수가 더 적은 작은 이더넷 LAN에서는 더 많은 소통량을 지원(충돌이 문제가 되기 전까지)한다. 충돌을 추적하여 언제 LAN을 세그먼트로 분리해야 혼잡이 줄어들고 성능이 개선되는지 파악할 수 있다.

(아) 토큰링 RMON 분석

토큰링 세그먼트의 상태 관련 데이터에는 토큰링 RMON 프로브에서 제공되는 매우 구체적인 또 다른 통계 데이터가 포함된다. 이 데이터를 보면 링 사용률, 오류 및 이벤트에 대해 자세히 알 수 있다. 또한 RMON은 어떤 종류의 오류가 링에서 일어나고 있는지 알려주는 기능도 가진다. 다음 〈표 4-26〉에서는 모니터 할 비율의 예를 보여준다.

〈표 4-26〉 토큰링 RMON 비율

	평균사용률(%)	최고사용률(%)	오류(%)	브로드캐스트/멀티케스트
경고 임계치	〉40	〉60	〉0.1	200
치명적 임계치	〉60	〉80	〉1	300

▶ 사용률 : 토큰링 사용률은 RMON 프로브를 통해 수집되며 링을 통과하는 소통량을 나타낸다. 사용률 패턴을 관찰하면 허용되는 수준 이내로 소통량 로드가 유지되도록 하고 해당 수준의 추세를 관찰할 수 있다. 이러한 추세를 통해 잠재적인 문제가 사용자들에게 영향을 미치기 몇 달 전에 그러한 문제를 파악할 수 있다.

평균 사용률 통계는 네트워크의 소통량이 어느 정도인지에 대한 정확한 시각을 제공한다. 그러나 짧은 기간 동안 활동이 폭증하는 현상은 나타내지 않는다. 그러한 폭증을 보여주는 것은 최고 사용량 통계이다. 최고 통계는 라우터와 직렬 회선의 크기를 조정하고 로컬 구성을 변경할 때 유용한 정보이다.

▶ 오류율 : 평균 오류율은 시간당 평균 오류율을 보여준다. 이러한 오류율은 물리 계층에 문제가 있음을 나타낸다. 이는 결함이 있는 케이블, 트랜시버 및 인터페이스 카드를 완전히 고장나기 전에 찾도록 도와준다. RMON 프로브에 대한 오류 정보를 보면 오류 유형을 구분할 수 있다.

제거, 오류 신호, 토큰 오류 등과 같은 발생하는 오류 유형을 파악하려면 추가로 오류를 분

석해야 한다. 또한 오류가 높은 사용률과 어떤 상관 관계를 보이는지 확인하기 위해 평균 및 최고 사용률 통계를 고려해야 한다. 그렇지 않다면 장애가 굳어지거나 네트워크가 오작동하기 전에 복원해야 하는 케이블 배선 오작동이나 하드웨어 결함이 있는 것일 수 있다

.

- 브로드캐스트/멀티캐스트 : 이 값은 초당 브로드캐스트/멀티캐스트 수를 나타낸다. 높은 비율의 브로드캐스트 소통량이 반드시 문제가 있음을 나타내는 것은 아니다. 값이 늘어나거나 갑작스럽게 피크에 도달하는 것은 LAN 하드웨어, 응용 프로그램 또는 계층 2 라우팅 문제(예: IPX SAP 폭풍) 등의 문제를 나타낼 수 있다. 이는 또한 대역폭을 불필요하게 사용하고 있다는 신호이기도 하다. 이 숫자는 링크 크기에 따라 다르다.

(자) FDDI RMON 분석

FDDI(Fiber Distributed Data Interface) 세그먼트와 관련된 데이터에는 FDDI RMON 프로브에서 제공하는 매우 구체적인 또 다른 통계 데이터가 포함된다. 이 데이터를 보면 링 사용률, 오류 및 이벤트에 대해 자세히 알 수 있다. 또한 RMON은 어떤 종류의 오류가 링에서 일어나고 있는지 알려주는 기능도 가진다. 〈표 4-27〉에서는 모니터 할 비율의 예를 보여준다.

〈표 4-27〉 FDDI RMON 임계 비율

	평균사용률(%)	최고사용률(%)	오류(%)	브로드캐스트/멀티케스트
경고 임계치	〉40	〉60	〉0.1	200
치명적 임계치	〉60	〉80	〉1	300

- 사용률 : FDDI 사용률은 RMON 프로브를 통해 수집되며 링을 통과하는 소통량을 보여준다. 사용률 패턴을 관찰하면 허용되는 수준 이내로 소통량 로드가 유지되도록 하고 해당 수준의 추세를 관찰할 수 있다. 이러한 추세를 통해 잠재적인 문제가 사용자들에게 영향을 미치기 몇 달 전에 그러한 문제를 파악할 수 있다.

평균 사용률 통계는 네트워크의 소통량이 어느 정도인지에 대한 정확한 시각을 제공한다. 그러나 짧은 기간 동안 활동이 폭증하는 현상은 나타내지 않는다. 그러한 폭증을 보여주는 것은 최고 사용량 통계이다. 최고 통계는 라우터와 직렬 회선의 크기를 조정하고 로컬 구성을 변경할 때 유용한 정보이다.

- 오류율 : 평균 오류율은 시간당 평균 오류율을 보여준다. 이러한 오류율은 물리 계층에 문제가 있음을 나타낸다. 이는 결함이 있는 케이블, 트랜시버 및 인터페이스 카드를 완전히 고장나기 전에 찾도록 도와준다. RMON 프로브에 대한 오류 정보를 보면 오류 유형을 구분할 수 있다.

  전환, 요구, CRC 오류 등과 같은 발생하는 오류 유형을 파악하려면 오류를 추가로 분석해야 한다. 또한 오류가 높은 사용률과 어떤 상관 관계를 보이는지 확인하기 위해 평균 및 최고 사용률 통계를 고려해야 한다. 그렇지 않다면 장애가 굳어지거나 네트워크가 오작동하기 전에 복원해야 하는 케이블 배선 오작동이나 하드웨어 결함이 있는 것일 수 있다.

- 브로드캐스트/멀티캐스트 : 이 값은 초당 브로드캐스트/멀티캐스트 수를 나타낸다. 높은 비율의 브로드캐스트 소통량이 반드시 문제가 있음을 나타내는 것은 아니다. 값이 늘어나거나 갑작스럽게 피크에 도달하는 것은 LAN 하드웨어, 응용 프로그램 또는 계층 2 라우팅 문제(예: IPX SAP 폭풍) 등의 문제를 나타낼 수 있다. 이는 또한 대역폭을 불필요하게 사용하고 있다는 신호이기도 하다. 이 숫자는 링크 크기에 따라 다르다.

(차) 가동시간 응답시간 분석

인터넷 제어 메시지 프로토콜(ICMP) 에코는 네트워크 리소스의 가용성을 테스트하기 위해 사용하는 테스트 패킷이다. 폴링 간격 동안 적은 수의 ping(일반적으로 5)을 보내서 하나라도 응답을 받으면 장치는 이 간격 동안 가동(사용 가능)되고 있다고 간주된다. 마찬가지로, 다양한 방법으로 두 개의 지정된 라우터/장치 사이에서 두 가지 목적으로 ping이 수행된다. 첫 번째 목적은 SLA 가용성 계산을 제공하는 것이고 두 번째 목적은 패킷의 네트워크 왕복 시간을 살펴보는 것이다.

▶ 가동 시간 : 가동 시간이 짧으면 장치나 장치로 연결되는 링크, 패킷을 포기하게 하는 링크 오버로드나 CPU에 대한 유지 관리를 포함한 다양한 동작이 발생할 수 있다. ICMP 패킷은 인터네트워킹 장치에 의한 실제 데이터 소통량과 비교할 때 우선 순위가 낮은 것으로 간주되므로 이 패킷이 제일 먼저 삭제된다. 그러나 여전히 이 패킷은 사용자가 네트워크의 특정 영역에 액세스할 수 있는지를 판단할 수 있는 척도이다.

▶ 응답 시간 : ICMP 에코를 주기적인 폴링 간격(예, 15분) 마다 보내어 이 값들을 평균하여 왕복 응답 시간을 계산한다. 응답 시간이 길다는 것은 일반적으로 네트워크, 특히 WAN 링크에서 혼잡이 있다는 것을 나타낸다. WAN 링크는 자주 막히는 지점이며 혼잡이 발생할 가능성이 크다. 사용률, 오류, 포기율 등의 다른 SNMP 관련 통계와 함께 이 데이터를 더 자세히 조사해야 한다. 이 값은 사용자가 네트워크의 특정 영역에 액세스할 수 있는지 판단할 수 있는 척도이다.

(카) 임계 요소가 아닌 요소 모니터링

▶ 패킷 크기 : RMON 통계 그룹 가능 장치에서 이 값을 보고 장치를 통과하는 패킷 크기를 알 수 있다. 일반적으로 다음과 같이 세분화할 수 있다.

- 64바이트 미만
- 65~127바이트
- 128~255바이트
- 256~511바이트
- 512~1023바이트
- 1024~1518바이트

▶ 호스트 카운트 : RMON 호스트 그룹 가능 장치에서 이 값을 보면 프로브가 연결된 특정 세그먼트에 의해 표시되는 장치 수를 알 수 있다. 현재 대부분의 인터네트워킹 장치는 그룹 4 RMON을 지원하지 않으며, 프로브를 통해서만 이러한 유형의 정보를 얻을 수 있다.

▶ 상위 송신기 및 수신기 : RMON 호스트 TopN(그룹 5) 가능 장치에서 이 값을 보면 이 프로브가 연결된 특정 세그먼트의 MAC 주소에 의해 상위 송신기를 알 수 있다. 현재 대부분의 인터네트워킹 장치는 그룹 5 RMON을 지원하지 않으며, 프로브를 통해서만 이러한 유형의 정보를 얻을 수 있다.

▶ 프로토콜 배포 : 일부 라우터는 라우터의 특정 인터페이스를 통과하는 프로토콜 유형을 알려준다. 이러한 몇 가지 프로토콜은 다음과 같다.

- AppleTalk, Bridge, DEC, IP, IPX, ISO, NetBIOS, STUN, Vines, XNS

▶ 응용 프로그램 계층 배포: RMON2 가능 장치에서 이 값을 보면 특정 세그먼트의 응용 프로그램 계층에서의 프로토콜 배포를 알 수 있다. 현재 대부분의 인터네트워킹 장치는 RMON2를 지원하지 않으며, 표준 기반의 RMON2 프로브를 통해서만 이러한 유형의 정보를 얻을 수 있다.

- FTP, Telnet, SMTP, NetBIOS, HTTP, NFS, IPX-NCP, IPX-SAP, DECNet, AppleTalk

(2) 측정 방법

측정 방법은 간단한 성능데이터의 경우 장비별 OS에서 제공하는 명령어 등을 이용할 수 있지만, 네트워크 관리 솔루션 제공사에서 제공하는 NMS(Network Management System) 도구를 이용하여 측정하는 것이 보다 풍부한 네트워크 성능정보를 손쉽게 분석할 수 있다. 특히 네트워크 성능 이력정보 분석을 위하여서는 NMS 도구를 이용하는 것이 좋다. 〈표 4-28〉에서는 측정 항목별로 영향을 미치는 요소 및 대표적인 측정 방법을 설명한다.

〈표 4-28〉 측정 항목별로 영향을 미치는 요소 및 측정 방법

측정 항목	영향 요소	측정 방법
가용성 (Availability)	▶ 가용성에 영향을 미치는 요소들 – 하드웨어 장애(Hardware Fault) – 소프트웨어 버그(Software bug) – 운영자의 실수(Human Error) – 전기적 문제(Electrical interference) ▶ 실제 환경에서 주의할 사항 – 장비 가용성(device availability) – 서비스 가용성(service availability)	▶ 측정 방법 가용성 = MTBF / MTBF + MTTR ▶ 실제 측정 방법 (NMS상에서 구현 방법) 가용성 = (수신 PING 개수 / 송신 PING 개수) × 100
응답시간 (Response Time)	▶ 응답시간에 영향을 미치는 요소들 – 네트워크 혼잡도(Network Congestion) – 라우팅 정보(Routing Information) – 네트워크 장비의 성능(Network Device Performance) – 회선의 장애(Noise, CRC)	▶ 응답시간 = (네트워크 Delay × 2) + 서버 Processing Time
정확성 (Accuracy)	▶ 정확성에 영향을 미치는 요소들 – 잘못된 케이블링(Out-of-specification wiring) – 전기적 문제(Electrical interference) – 하드웨어 장애(Hardware Fault)	▶ Error Rate = (@ifInErros × 100) / (@ifInUcastPkts + @ifInNUcastPkts) ▶ Accuracy = 100 – ((@ifInErrors) × 100) / (@ifInUcastPkts + @ifInNUcastPkts))

사용률 (Utilization)	▶ 네트워크 자원을 일정시간 사용하는 정도	▶ Input Utilization = (@ifInOctets × 8 × 1000) / ((number of seconds in @) × @ifSpeed) ▶ Output Utilization = (@ifOutOctets × 8 × 100) / ((number of seconds in@) × ifSpeed)

〈표 4-28〉의 측정 방법은 다음 설명을 참조하여 분석한다.

▶ MTBF = (Mean Time Between Failure)
인시던트가 복구된 시점부터 다음 인시던트가 보고된 시점 사이의 평균 시간, 즉 Uptime을 말한다.

▶ MTTR = (Mean Time To Repair)
문제가 발생한 시점부터 복구한 시점까지의 평균 시간, 즉 Downtime을 말한다.

▶ @ifInErrors = 전송되어 들어온(inbound) 에러 패킷의 개수

▶ @ifInUcastPkts = 전송되어 들어온(inbound) 유티캐스트 패킷 개수

▶ @ifInNUcastPkts = 전송되어 들어온 (inbound) 비-유티캐스트패킷 개수

▶ @ifInOctets = 전송되어 들어온(inbound) 트래픽의 Octets 값

▶ @ifOutOctets = 전송되어 나간(outbound) 트래픽의 Octets 값

▶ @IfSpeed = Interface의 Speed 값

(3) 분석 방법

다음과 같은 성능측정 방법을 통하여 네트워크 성능분석을 수행한다.

▶ PING : 특정 IP Address에 접근 가능한지 체크한다. 즉, IP address를 가진 시스템이 네트워크에 정상적으로 접속되어 있는지 여부와 내 시스템과 상대 시스템간의 지연상태와 품질상태를 알 수 있다.

ex) ping hoss1.interpia98.net

ping 210.124.122.97

▶ PING 응답시간 : 응답시간은 패킷이 목적지까지 갔다가 돌아오는 시간을 측정한 시간이며 단위가 ms(mili second)로서 전용회선 속도에 비례하며 그시간이 짧을수록 좋지만 테스트 당시의 전체 네트워크에 걸리는 부하에 따라서 조금씩 차이가 날수 있다. 'request timed out' 이라는 메시지가 나타나면 어느 구간이 불안정한 상태이다. 이 경우 다음에서 기술한 순서대로 구간점검을 하여야 한다.

- PC및 서버 네트워크 환경설정이 정상인가?
- 랜카드 및 허브구간은 정상인가?
- 게이트웨이까지 ping 응답은 정상인가?
- 네트워크 서비스 제공자의 라우터 시리얼 IP까지 ping은 정상인가?
- 네임서버까지 정상인가?

위 점검단계에서 3번째 단계까지 이상이 발견되지 않으면 네트워크 서비스 제공자에게 장애신고를 하여 해결한다.

▶ SNMP(Simple Network Management Protocol) : 네트워크 장비를 모니터링 및 제어하고 장애, 성능 관련 통계 수집 및 보안등의 관리를 위한 프로토콜이다. SNMP를 이용해서 할 수 있는 것들은 다음과 같다.

- 네트워크 구성 관리 : 네트워크상의 호스트들이 어떤 구조를 이루고 있는지 지도를 그리는게 가능하다.
- 성능 관리 : 각 네트워크 세그먼트간 네트워크 사용량, 에러량, 처리속도, 응답시간 등 성능 분석에 필요한 통계정보를 얻어낼 수 있다.
- 장비 관리 : SNMP의 주목적이 네트워크 관리이기는 하지만 SNMP특유의 유연한 확장성을 이용하여서 시스템(CPU, 메모리, 디스크 사용량)의 정보를 얻어올 수 있도록 많은

부분이 확장되었다. 이 정보는 네트워크 문제를 해결하는데 큰 도움을 준다. 예를 들어, 특정 세그먼트의 네트워크 사용량이 갑자기 급증했는데, 특정 호스트의 CPU 사용율까지 갑자기 증가했다면, 우리는 해당 호스트에서 문제가 발생했을 것이라고 유추해 낼 수 있을 것이다.

- 보안 관리 : 정보의 제어 및 보호 기능이 있으며, 최근 버전인 SNMP3는 특히 정보보호를 위한 기능이 향상되었다.

▶ SNMP 통신 : Agent(Server) Manager(Client) – 네트워크 성능관리 도구 이용

- Agent : 각 네트워크 노드 내에서 돌아가는 프로그램
- Manager : 네트워크 모니터링 단말에서 운영되는 소프트웨어(NMS)

▶ 패킷 캡쳐어 (Packet Capture) – 네트워크 성능관리 도구 이용

- Packet Capture
- LAN/WAN segment상에서 직접 raw packet 정보를 통해 패킷 정보 수집
- L2 Frame에서부터 L7 데이터 정보까지 폭넓은 정보 수집 가능
- LAN Switching 환경에서는 포트 미러링(Port mirroring 또는 SPAN) 사용
- 고속, 고용량 대역폭에서 패킷 캡쳐링 수집의 한계

나. 성능개선 방안 수립

(1) 대상 선정

공공부문의 운영파트와 개발파트는 성능분석 결과를 바탕으로 성능개선 항목이 있는지 도출하여 해당 사항에 대한 개선대책을 수립한다.

〈표 4-29〉 네트워크 성능 개선 대상 선정

구분	성능 개선 대상 선정
실시간 성능 문제 (Real-time Network Performance Problem)	▶ 성능 문제 현상 - 갑작스럽게 인터넷 접속(WWW)이 느려진다. - Ping은 되는데, 메일 접속이 안된다.
비 실시간 성능 문제 (non Real-time Network Performance Problem)	▶ 성능 문제 현상 - 지난 주에는 괜찮았는데 오늘 아침에는 메일이 너무 느리다. - 주기적으로 인터넷 접속이 느리다.

성능개선 항목 선정 시 다음과 같은 대책을 고려해야 한다.

- Configuration 재구성
- 사용율 초과로 인한 전송속도 및 대역폭 증설
- 장비 에러로 인한 장비 교체
- 장비별 MIB 수정

(2) 방안 수립

도출된 성능개선 사항이 네트워크와 관련된 경우 운영파트는 관련 성능개선 대책 보고서를 작성하여 운영파트 PM에게 보고 후 승인을 받는다. 운영파트 PM은 성능개선 대책의 사안에 따라 IT 팀장에게 보고 후 승인을 받는다.

도출된 성능개선 사항이 어플리케이션과 관련된 경우 개발파트는 관련 성능개선 대책 보고서를 작성하여 IT 팀장에게 보고 후 승인을 받는다. 네트워크의 성능을 개선해야 할 대상을 선정하였으면 다음 〈표 4-30〉에서 언급한 바와 같은 방법으로 해결방안을 수립하여 단계별로 해결하여 나간다.

〈표 4-30〉 네트워크 성능 문제 해결방안

구분	해결방안
실시간 성능 문제 (Real-time Network Performance Problem)	▶ 단계별 성능 취약점 분석을 통한 성능 문제점 해결 ▶ 일시적인 네트워크 성능 문제 현상으로써 네트워크 장애(Fault)와 연관됨
비 실시간 성능 문제 (non Real-time Network Performance Problem)	▶ 기존의 네트워크 자료를 통해 접근해야 함 ▶ 성능 문제를 판단하기 위한 Base-line 작업이 필요 ▶ 네트워크 용량(Capacity)과 연관됨

□ 네트워크 성능 문제 해결 단계

① 소스(source)와 목적지(destination)를 확인

▶ 성능 장애가 발생한 소스와 목적지에 대한 정확한 IP주소 확인

▶ 방법

– 단순하지만 간과하기 쉬운 요소 → 가급적이면 MAC주소까지 확인 필요

② 서버의 트래픽 사용량(traffic usage)을 확인

▶ 서버의 CPU와 NIC 사용률(Utilization)을 확인할 것

▶ 방법

– host에서 실행 (작업관리자, top, ifconfig)

예) Unix 계열 : top, ifconfig [interface]

Windows 계열 : 작업관리자 – 성능

③ 소스와 목적지간의 경로(path)를 확인

▶ 방법

- L3 정보(라우팅 정보)
- L2 정보(스위치 정보, Spanning-Tree Protocol)

④ 경로의 각 구간별 사용률(utilization)을 확인

▶ 방법

- 라우터/스위치에서 명령어(show interface, show port) 확인
  - 라우터의 interface의 사용량 및 장애 현상을 명령어로 확인할 것
    예) show interface fastethernet 1/0
  - 스위치의 port의 사용량 및 장애 현상을 명령어로 확인할 것
    예) show port 1/1
- NMS를 통한 트래픽 현황 확인(성능관리도구 이용)
  - 실시간 정보 및 히스토리칼 정보(성능추이 정보)

⑤ 네트워크 장비의 성능(CPU, 메모리, 버퍼)을 확인

▶ 방법

- 라우터에서의 CPU 사용률 확인 및 IP traffic 확인
  예1) show processes
  예2) show ip traffic
- 라우터의 Memory와 buffer 상태를 확인
  예1) memory summary
  예2) show buffer

⑥ Raw Packet을 분석 확인

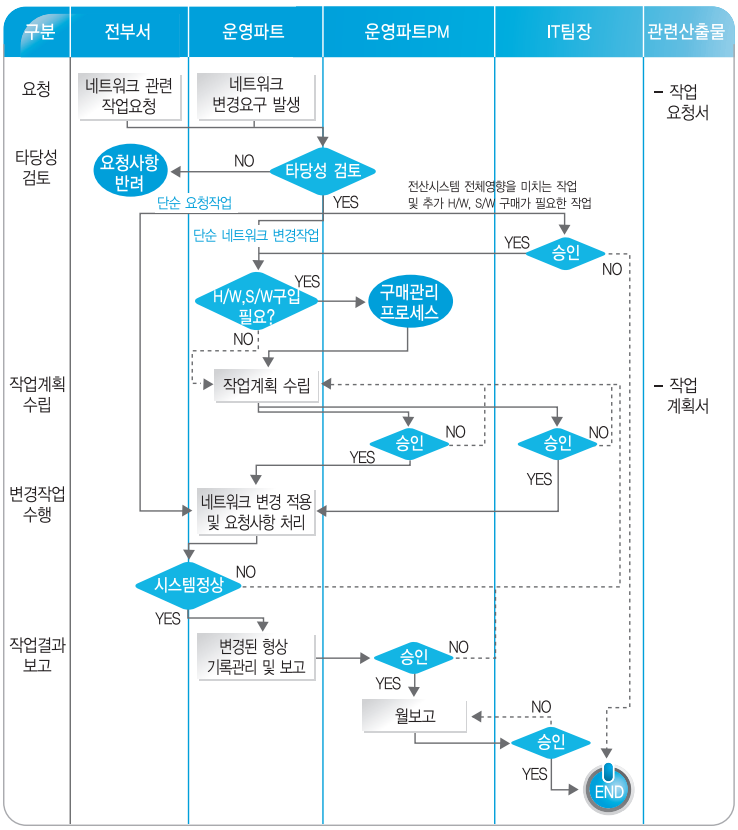
▶ 방법

– Network Analyzer를 통해 어플리케이션 분석

다. 성능개선 실행 및 검증

(1) 성능조정(튜닝)

(가) 성능조정 항목이 시스템 관련 사항인 경우 (그림 4-7)의 '네트워크 변경적용 및 요청작업 처리' 에 따라 성능개선 대책을 적용한다.



(그림 4-7) 네트워크 변경작업 및 요청작업 처리

(나) 성능개선 항목이 응용소프트웨어 관련 사항인 경우 본 지침의 '응용 소프트웨어' 부분에 따라 성능개선 대책을 적용한다.

(다) 성능조정 항목이 단순 사항인 경우 운영 파트 네트워크 관리자는 요청작업을 처리하여 요청 파트 및 팀으로 통보한다. 요청 파트 및 팀에서는 요청작업 결과를 확인한다.

(라) 성능조정 항목이 네트워크 변경 작업인 경우 운영 파트 네트워크 관리자는 해당 작업이 운영 시스템에 미치는 영향(하드웨어, OS, DBMS, 어플리케이션, 시스템 관리도구 등에 대한 영향과 파급효과) 및 작업내용의 적정성을 판단 후 적용 여부를 결정한다. 운영 파트 네트워크 관리자는 요청작업 적용여부를 판단한 후 운영 파트 PM에게 보고 후 결과를 작업요청 파트 및 팀에게 통보한다.

(마) 운영 파트 자체 사업계획 및 네트워크 성능분석 결과 네트워크 구축/변경이 필요한 경우 해당 작업이 운영 시스템에 미치는 영향 및 작업의 타당성을 검토한다. 운영 파트 PM은 자체 계획에 의한 작업이 전체 전산시스템에 영향을 미치는 경우 또는 별도의 H/W 및 S/W의 도입이 필요한 경우 IT 팀장에게 작업내용을 보고하여 승인을 득한다. 작업계획의 확인이 완료되면 운영 파트 네트워크 관리자는 작업 계획서에 따라 네트워크 변경사항을 적용시킨 후 네트워크 및 시스템 정상유무를 개발 파트 담당자와 확인한다. 네트워크 및 시스템 정상유무 확인은 어플리케이션과 OS, DB 등의 주요 시스템 기능을 사전에 작성된 점검표를 기준으로 확인한다.

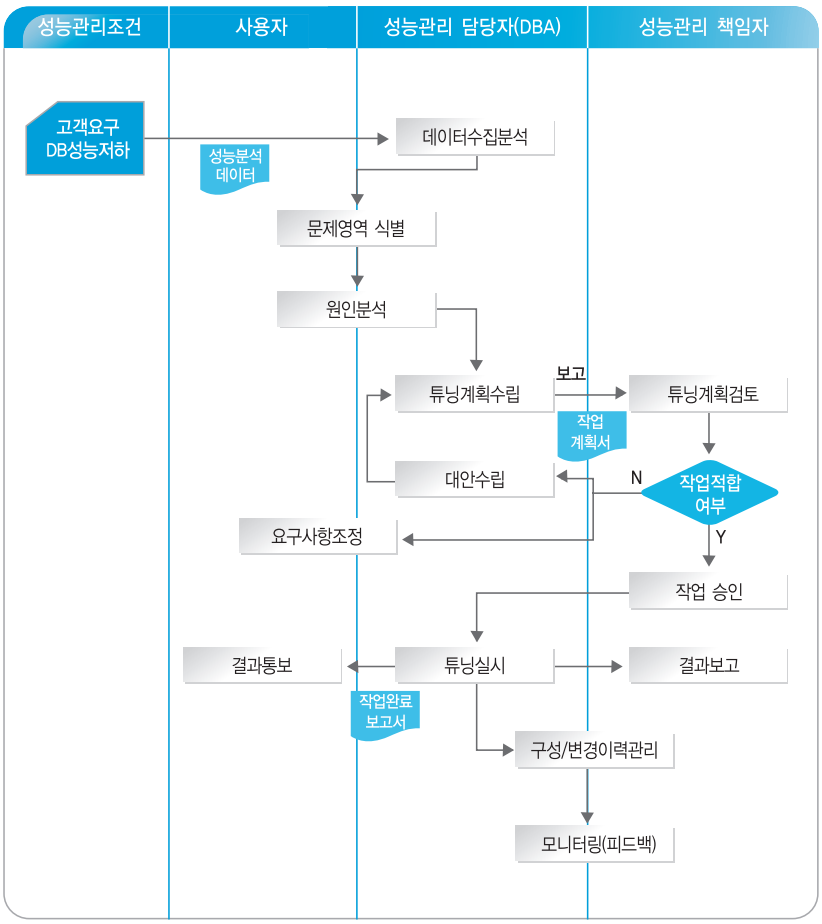
(2) 용량 증설

전산기 가용성 증대를 위한 네트워크 용량증설 및 네트워크 관련 작업 필요 시 작업요청 파트 및 팀은 관련 내용을 정형화된 양식(본 지침의 부록 참조 – 용량증설 요청서)을 작성하여 담당 관리자의 승인을 득하여 운영 파트로 요청한다.

운영 파트 자체 계획 및 시스템 성능분석 결과 네트워크 증설/변경이 필요한 경우 운영 파트 자체적인 타당성 검토를 수행한 후 해당 사안이 전산시스템 전체에 영향을 미치는 경우, 또는 IT 팀장의 승인이 필요한 경우 IT 팀장의 승인을 득한다.

4 2.3 DBMS

DBMS 성능관리 프로세스는 (그림 4-8)과 같은 흐름으로 진행된다.



(그림 4-8) DBMS 성능관리 프로세스

① 성능관리 개시조건 : 고객의 성능개선 요구 및 데이터베이스의 성능저하 시

② 데이터 수집 및 분석 : CPU, 메모리, 디스크 사용량 및 데이터베이스 사용량, DBMS 관련 네트워크 사용량 등 성능분석 데이터를 수집하고 분석한다.

③ 문제영역 식별 및 원인분석 : 특정 프로그램이나 특정 자원의 사용량 등의 특이사항을 분석하고, 상호작용을 파악하여 문제영역을 식별하고, 그 원인을 분석한다.

④ 성능 조정(튜닝) 계획 수립 및 보고 : 분석된 결과를 근거로 하여 성능 조정(튜닝) 계획을 수립하고 담당 관리자에게 해당사항을 보고한다.

⑤ 성능 조정(튜닝) 작업계획 검토 및 적합여부 결정 : 성능 조정(튜닝) 작업계획서는 성능 관리 책임자의 검토 및 작업승인을 받아야 한다. 계획이 부적절한 경우 대안을 수립하거나, 요구사항을 조정하고 다시 작업계획서를 작성하여 승인을 받는다.

⑥ 성능 조정(튜닝) 작업 실시 및 보고 : 조정(튜닝) 작업 실시 후, 작업의 정상완료 여부를 작업완료 보고서에 작성하여 성능관리 책임자에게 보고하고 성능분석 요청(의뢰)자에게 결과를 통보한다.

⑦ 변경이력 관리 및 모니터링 : 조정(튜닝) 작업 결과 변경 사항에 대한 이력을 갱신하고 DBMS를 지속적으로 모니터링 한다.

가. 데이터 수집 및 분석

(1) 측정 항목

DBMS 성능관리의 대상은 일반적으로 스키마(Schema), SQL 문장/저장 프로시저, 공유 메모리 영역, 데이터베이스 파일 관리, 세그먼트 관리, 정렬 영역, 롤백 세그먼트, Locking 관리, 세션 관리 영역으로 분류할 수 있으며, 각 대상별로 세부적인 측정항목을 선정한다. 이러한 대상 분류는 운영상태관리 분야의 대상과 중복되는 부분도 있으나, DBMS의 성능(Performance)을 대변하는 '응답시간 (Response Time)' 및 '처리량(Throughput)'과 관련된 항목을 세부 측정 항목으로 선정하는 것이 일반적이다.

〈표 4-31〉 성능 측정 대상별 측정 항목 및 주기

대 상	측정항목	측정주기
스키마 (schema)	- 과도한 정규화(Normalization)로 인한 조인의 발생 (속도 저하 발생)	구축시
	- 과도한 반정규화(De-Normalization)로 인한 중복 데이터의 발생 (공간 낭비 발생)	구축시
	- 인덱스 부재로 인한 Full Table Scan의 발생 (과도한 I/O 발생)	구축시
	- 비정상적으로 작성된 인덱스의 사용	월간/ 비정규적
	- 사용되지 않는 인덱스	
응용프로그램 (SQL, 저장 프로시져)	- 80/20 법칙을 적용하여 Top SQL 문장을 선정	주간
	- Full Table Scan을 발생시키는 SQL 문장 (과도한 Disk I/O를 유발하는 SQL 문장)	주간
	- CPU 자원을 과도하게 사용하는 SQL 문장	매일
	- Buffer Cache를 과도하게 사용하는 SQL 문장	매일
	- OLTP 업무 수행 시간대에 실행되는 배치성 SQL 문장	매일
공유메모리 영역	- 공유 메모리 내 Buffer Cache의 Read/Write/Update/Free 상태 검사	매일
	- Database Object중 Shared Pool에 Cache된 Object 검사 (Trigger, Procedure, Package, Package Body, SQL)	주간
	- 공유 메모리 내 Data Buffer Cache의 적중률을 측정	매일
	- 공유 메모리 내 Shared Pool에 Cache된 SQL TEXT 검사	주간
	- 자료사전(Data Dictionary) 정보 관련 Parameter 통계 정보	주간
	- 데이터 블록 경합(Contention) 통계정보	매일
	- 공유 메모리 사용 요약 정보	주간
	- SQL 문장의 Parsing 영역에 대한 Misses Ratio 정보	주간
	- Redo Log Buffer에 대한 대기(Wait) 횟수 검사	매일
데이터베이스 파일 관리	- 데이터 파일의 분산배치	구축시
	- 컨트롤 파일의 분산배치 및 이중화	구축시
	- 리두로그 파일의 분산배치 및 이중화	구축시
	- 아카이브 모드의 데이터베이스 운용	구축시/운영중
	- 특정 datafile을 이용하고 있는 oracle user에 관한 검사	매일
	- 특정 datafile에 포함된 segment 검사	매일
	- Database를 구성하고 있는 datafile에 관한 정보 조사	월간
	- 특정 크기보다 큰 디스크 사이즈를 가진 Tablespace 목록	월간
	- Tablespace의 여유 공간 검사	주간
	- Tablespace의 상황 검사	주간
	- Tablespace에서 연속된 공간의 Size를 검사	주간
	- Tablespace를 구성하는 datafile 검사	월간
	- datafile의 상태 검사	월간

세그먼트 관리	- 특정 테이블에 할당된 Block수와 실제 사용하고 있는 Block의 수 - Index의 storage parameter 검사 - Extent 숫자 ,크기에 따른 분류 - 특정 Table의 storage parameter 검사 - 세그먼트(테이블, 인덱스 등)의 단편화(Fragmentation) 관리 - 한 Block에 있는 최소, 최대, 평균 Row의 수 - Table의 storage parameter 검사 - 특정 Oracle User가 가지고 있는 Table 및 Parameter 조사 - 특정 테이블에서 data를 가지고 있는 block의 수 - data file에 있는 특정 segment을 검사 - database 내에서 가장 size가 큰 segment의 검사 - table의 extent 수 검사	주간 생성시/비정규적 생성시/비정규적 생성시/비정규적 주간 월간 생성시/비정규적 월간 주간 월간 월간 주간
정렬영역	- 과도한 정렬 작업을 발생시키는 프로세스 - 과도한 정렬 작업을 발생시키는 SQL 문장의 판단 - 메모리 정렬 영역 및 매개변수 조정 여부 - 메모리 정렬 대비 디스크 정렬 작업의 비율 - 임시 영역 세그먼트 할당 및 할당 해제 - 정렬(Sort) 통계 정보	매일 매일 주간 주간 주간 월간
롤백세그먼트 관리	- 롤백 세그먼트의 대기(Wait) 이벤트 - 블로킹(Blocking) 세션(Session)의 감시 - 트랜잭션 당 적절한 롤백 세그먼트 크기 조정/관리 - 롤백 세그먼트 내의 트랜잭션 정보 - 롤백 세그먼트의 저장영역 파라미터 - 롤백 세그먼트의 경합(Contention) Check 및 Stats Check	매일 매일 주간 매일 생성시 매일
Locking 관리	- Deadlock 정보 검사 - Database에 발생하고 있는 모든 Lock 정보 - Database내의 Lock 관련정보와 Object정보 - Redo Log Buffer Latch Contention Check - 모든 Latch Contention 정보 - Session이 소유하고 있는 Lock 정보	매일 매일 매일 주간 주간 매일
유저(세션) 관리	- User당 Buffer Hit Ratio가 80%이하인 User 정보 - Process Idle Time Check - 가장 CPU를 많이 사용하고 있는 Process 정보 - Process,Session,Client,Idle Time,OS User정보 - 특정 Cursor에서 실행중인 SQL 정보 - 특정 Tablespace에 속한User,사용된 Space Quota정보 - 현재 Transaction이 Active한Session,Process,OS User정보 - Database에 생성된 User 정보 - Process가 수행하고 있는Transaction정보 (SQL 정보) - Session이 수행하고 있는Transaction 정보 및 Undo Size	매일 매일 매일 주간 비정규적 월간 매일 월간 비정규적 비정규적

DBMS 성능관리의 세부 측정항목은 운영상태관리의 모니터링 항목과 중복되는 항목들이 많으나, 구분하는 기준은 사용자의 SLA 및 성능 만족도를 충족시키기 위한 응답시간(Response Time), 처리량 (Throuthput), 자원 사용량(Utilization), 효율성(Efficiency)을 판단하여 성능관리 측정 대상 항목을 결정한다. 따라서 일상적인 고장 장애를 모니터링하기 위한 항목은 성능관리 측정 항목에 포함시킬 필요가 없다.

(2) 측정 방법

(가) DBMS 측정 항목별 임계치 설정

각 측정 항목의 임계치는 SLA를 만족하는 기준에 의하여 설정되며, 사용자의 만족도와 추가적인 요구사항을 고려하여 설정한다. 측정 세부 항목별 임계치는 DBMS 제작사 (Vendor)에서 제공하는 가이드라인을 따르는 것이 일반적이지만, 업무 역할(Role)에 상응하는 기관의 DBMS 활용 특성이 반영되어야 한다.

예를 들어, 오라클 RDBMS의 데이터베이스 버퍼 캐쉬 적중률(Hit Ratio)은 업무 형태에 따라, OLTP 업무의 경우 90% 후반을, DSS(혹은 DW)업무의 경우 60~70% 이상을 유지하면 성능이 안정적이라고 판단한다. 따라서, 성능 임계치를 고정된 하나의 기준으로 설정하는 것보다, 업무 형태와 특성을 고려하여 사용자의 응답시간(Response Time), 만족도 등을 적극적으로 반영하여야 한다.

▶ 임계치 설정 시 반영(고려) 요소

- DBMS 제작사의 성능 판단 기준 자료 및 권고사항
- SLA 기준
- DBMS 활용 형태(OLTP, DSS/DW, OLAP 등)
- 사용자의 만족도(응답시간, 처리량) 평가 결과
- 운영상태관리 분야로부터의 피드백

(나) DBMS 성능 측정 및 분석 주기

DBMS 성능분석은 일반적으로 일간, 주간, 월간 주기로 측정 대상을 분류하여 성능데이터의 수집과 분석을 진행하며, DBMS를 이용하는 비즈니스 순환주기(Business Cycle)에 그 근간을 둔다.

▶ 일간 성능 측정 및 분석

- 각 관리 항목별 시간대별 성능 추이를 조사, 정리하며 전일 대비 특이 사항이 없는지 점검한다.
- 일별 성능 데이터는 월간 성능 데이터에 누적되어 해당 월의 항목별 성능 추이를 점검하도록 한다.

▶ 주간 성능 측정 및 분석

- 주간 단위 성능 측정 및 분석은 필수 분석 주기는 아니다. 그러나 업무 특성상 일주일 주기로 발생하는 트랜잭션에 대하여 성능 분석 요구가 발생할 경우 실시한다. 예를 들어, 주 단위로 DBMS의 백업을 실시할 경우, 성능 장애로 인하여 제시간에 백업이 이루어지지 않아 업무시간 대까지 진행되는 경우 등에 한하여 데이터베이스 백업 오퍼레이션에 대한 성능 측정 및 분석을 수행할 수 있다.
- 필요한 경우, 주간 성능 데이터 역시 누적되어 월간 성능 추이 분석에 활용할 수 있다.

▶ 월간 성능 측정 및 분석

- 일반적으로 월간 성능 측정 및 분석은 업무 시스템 단위로 적용하여 전/후 1주일간의 성능데이터를 수집, 분석하여 보고서를 작성한다.
- 주간 성능 측정 및 분석 데이터를 유지하는 경우에는 4주간의 누적치를 합산하여 월간 성능 추이를 분석한다.
- 성능지표에 대한 월간 성능변화 분석 성능지표에 해당 하는 항목의 성능데이터를 수집하여 분석 보고서를 작성한다.

▶ 비정규적인 성능 측정 및 분석 주기

- 정기적인 성능 측정 및 분석 외에 SLA 조건을 위반하는(임계치를 초과하는) 현상이 발생하거나, 사용자의 긴급한 성능 분석 요구가 발생할 경우 등에 한하여 비정규적인 성능 측정 및 분석이 이루어 질 수 있다.
- 비정규적으로 발생하는 성능 측정 및 분석 결과는 정규적으로 실시하는 성능 측정 및 분석에 반영하지 않는다.

(3) 분석방법

분석 항목은 운영 환경 및 성능 측정 환경의 특성에 따라 조정 가능하며, 일반적으로 DBMS 메모리 분야, 데이터의 저장영역, 어플리케이션 지향적인 SQL 문장 및 Locking 관련된 분야로 크게 나누어 분석한다. 수집된 정보를 다음과 같은 항목으로 분석한다.

(가) 분야별 성능 분석 방법

▶ DBMS 메모리 성능 분석

- 공유 메모리 영역의 성능 분석은 대부분 적중률(Hit%)을 기준으로 이루어진다. 적중률이란 프로세스가 원하는 데이터가 공유 메모리에 존재하여 추가적인 디스크 I/O가 발생하지 않는 것을 의미한다. 만일 공유 메모리 영역에 존재하지 않아서, 디스크 상의 데이터 파일로부터 데이터를 가져와서 공유메모리에 캐싱한 후 이 데이터를 프로세스에게 공급한 경우를 Miss 하였다고 하며, 100%에서 적중률(Hit%)을 제외한 %를 말한다.
- 메모리 영역의 성능 분석은 SQL 문장의 Parsing이 발생하는 Shared Pool 영역, 데이터가 캐싱되는 Database Buffer Cache영역, 트랜잭션의 리두 정보가 처리되는 Redo log Buffer의 성능 상태 분석으로 크게 이루어진다. 이 외에도 다양한 메모리 성능 분야가 존재하며 제작사별로 상이한 통계값과 임계치(Threshold)를 가진다.

▶ 저장영역 관련 성능분석

- 데이터베이스의 세그먼트들이 저장된 영역의 분산 배치 및 이중화는 데이터베이스가 I/O 지향적인(I/O intensive) 작업이라는 특징을 고려하였을 때, I/O 경합(Contention)의 분석이 가장 우선되어야 한다. 성능분석 시 기준이 되는 고려사항은 아래와 같다.
  - 요청이 많은 OLTP 응용 프로그램의 경우, Dictionary로 관리되는 테이블스페이스를 사용할 때는 확장 영역을 할당하는 동안 영역 관리 작업을 위해 Dictionary에 액세스하는 등의 동시성 문제가 발생하는지 여부를 고려해야 한다.
  - 사용자 생성 시 사용자에게 필요한 모든 디스크 정렬과 임시 테이블 생성을 위해 임시 테이블스페이스가 할당되어야만 한다. 이러한 정렬 영역의 다른 데이터베이스 객체와 구분되는지 여부를 고려해야 한다.
  - 테이블과 인덱스는 동시에 삽입되고 읽혀지는 경우가 많으므로 개별 테이블스페이스로 분할되어야 한다. 따라서, 데이터가 저장되는 테이블스페이스와 인덱스가 저장되는 테이블스페이스가 분리되어 있는지 여부를 고려해야 한다.
- 결정적으로 모든 데이터베이스 파일은 별도의 Disk Array 장비에 배치하여 H/W가 제공하는 I/O 분산 효과를 이용하고 있는지를 운영체제 레벨에서 모니터링 및 분석하여야 한다.

▶ SQL 문장 성능 분석

- SQL 문장을 작성하는 개발자는 문장의 Performance를 고려하여 작성하는 것보다는 SQL의 결과값에 관심을 가지는 것이 일반적이다. 따라서, DBMS 성능관리 담당자(DBA)는 항상 시스템의 TOP SQL 문장을 추출하여 그 Plan을 분석하여야만 한다.
- 잘못된 Plan, 또는 비효율적인 Access Path를 가지고 있는 SQL 문장에 대해서는 개발자와 상의하여 적적한 Plan을 갖도록 대체 SQL 문장을 구사하거나, 적절한 Access Path를 확보하도록 새로운 Index 정책을 수립하여야 한다.
- 따라서, SQL 문장 및 저장 프로시저(Stored Procedure) 내의 SQL에 대해 주기적으로 그 Plan을 검사하여 분석하는 것은 데이터베이스 성능 관리의 핵심 오퍼레이션으로 볼 수 있다.
- SQL 문장의 분석은 운영체제의 Top Process의 분석과도 밀접하게 관련되어 있으므로 해당 운영체제에 대한 분석이 병행하여야 한다.

▶ Locking 관련 성능분석

- Locking 관련 이슈는 일반적으로 어플리케이션 로직 상의 오류와 밀접하게 관련되어 있다. 데이터베이스 서버는 잠금을 자동으로 관리한다. 기본 잠금 방식은 최고의 데이터 동시성을 허용하면서 데이터의 일관성을 보장할 수 있도록 최하위 제한 레벨(로우 레벨)에서 데이터를 잠근다.
- 따라서, Performance Dictionary View를 통하여 발견된 Locking의 대부분은 Application의 Hang과 연관되어 있을 가능성이 높으며, 따라서 Application의 Locking Algorithm을 통하여 분석 및 개선될 수 있다.
- 특히 DeadLock과 같은 교착상태를 초래할 경우 DBMS는 심각한 성능상의 문제를 발생하게 되므로 성능관리 담당자는 Deadlock의 모니터링 및 분석에 많은 관심을 기울여야 한다.
- Locking 관련 분석은 일반적으로 Lock Holder과 Lock Waiter로 분리되며, 세부적인 SQL 및 세션 트레이스를 통하여 해당 SQL 문장의 자세한 Activity를 판단할 수 있다.

(나) 항목별 성능 추이 및 성능 수준을 분석

▶ 개별적인 측정 항목별 성능 데이터의 기록을 누적한다.

▶ 누적된 개별 데이터는 추이 분석을 통하여 일별, 주간별, 월별 성능 상태를 분석한다.

▶ 각각의 측정 항목별 분석 기록을 기반으로 DBMS 전반의 통합적인 성능분석을 위하여 종합 분석한다.

(다) 성능 측정 결과를 이용한 성능 분석 주기

▶ 성능 측정 결과 안정된 성능 상태를 보이는 측정항목에 대하여는 성능 분석 주기를 조절할 수 있다. 일간주기로 측정되던 항목이 수개월간의 분석 결과 안정화된 성능 상태를 지속적으로 보이는 경우, 이 항목의 측정 주기는 주간 또는 월간 단위로 변경할 수 있다.

- 일간주기로 실시하는 분석 방법은 실시간 또는 5~10분 이하의 짧은 간격으로 측정하여 즉각적인 성능상태를 모니터링하고 분석할 수 있는 데이터 수집을 목적으로 한다. 이 경우 데이터의 수집이 시스템의 과부하(Overhead)로서 작용할 가능성이 있으므로 주의하여야 하며, 만일 이러한 성능 데이터 수집 항목이 안정화된 성능상태를 지속적으로 나타내거나, 성능상태가 변경될 가능성이 거의 없을 경우는 주간, 또는 월간 주기로 변경할 수 있다.

(라) 계층별 성능분석

데이터베이스 시스템의 성능관리 대상을 종합분석의 관점에서 볼 때는 일반적으로 다음의 세 가지 계층으로 구분한다.

- O/S Layer : CPU, 메모리, 디스크 IO, 네트워크 경합 해소 등
- DBMS Layer : 메모리 및 자원 할당, 저장 공간 설정(객체의 특성 고려), 데이터 파티션, 파일 I/O 분산 …
- Application Layer : 비즈니스 규칙, 모델링, 프로그램 로직, SQL 등

계층별 성능분석 및 튜닝을 수행할 때에도 응답시간과 어플리케이션의 관점에서 접근하는 것이 바람직하다. 데이터베이스 혹은 O/S에서의 자원사용 효율성은 근본적으로 작업을 요청하는 어플리케이션에 의해 좌우되는 것이므로, 특정 계층의 몇몇 통계지표만을 부분적으로 분석해서는 근본적인 원인을 찾는다거나 구체적인 성능조정(튜닝) 행위를 수행하기가 어려워진다.

응답시간의 추이를 모니터링하고 분석하기 위해서는 시중에 판매되는 성능 모니터링 전문 도구를 사용할 수도 있겠으나 기본적으로 데이터베이스에서 제공되는 성능관련 통계 정보들, 특히 대기 시간과 관련된 정보들을 주기적으로 수집하고 모니터링 함으로써 필요한 정보를 얻어낼 수 있다. 응답시간은 결국 실제로 시스템 자원을 소모하면서 작업을 처리한 서비스 시간과 시스템 혹은 데이터베이스 자원을 획득하기 위하여 소모한 대기 시간으로 구성된다고 할 수 있다. 이때, 서

비스 시간은 데이터베이스가 제공하는 CPU 시간을 통해서 알 수 있으며 대기 시간은 각종 대기 이벤트에 대한 통계치를 분석하여 알아낼 수 있다.

전체 데이터베이스 시스템, 세션, SQL 단위로 처리시간과 대기시간의 비중을 분석하여 만약 서비스 시간의 비중이 매우 크다면, 서비스 시간을 많이 사용한 SQL이 어느 것인지 추출하여 튜닝을 통해 개선할 요소가 없는지 찾아내야 한다. 만약, 대기시간의 비중이 서비스 시간에 비하여 매우 크다면 많은 비중을 차지하는 대기 이벤트부터 상세하게 원인 분석을 수행해야 한다. 대기 현상을 분석할 때에는 해당 대기 이벤트가 O/S 및 시스템 자원에 대한 것인지 데이터베이스 자원에 대한 것인지를 밝혀야 하는데 이를 위해서는 데이터베이스와 O/S의 성능지표를 함께 살펴야만 한다.

〈표 4-32〉 성능 측정 항목(요소)에 대한 측정 및 분석 방법

측정 항목	측정요소	측정/분석 방법
스키마 (schema)	- 정규화(Normalization) 요소 - 인덱스 정책 - 테이블 스토리지	- 정규화는 최초 구축 및 세그먼트 생성시 결정되는 요소이며, 운영 중에는 이로 인하여 발생하는 과도한 자원의 사용을 모니터링하여야 한다. - 인덱스의 적절성 여부는 SQL 문장의 Plan을 모니터링 및 분석한다. - 테이블의 잘못된 스토리지 설정은 과도한 Extent의 할당으로 나타나게 되므로 Performance Dictionary View의 Storage관련 view를 모니터링한다.
응용프로그램 (SQL, 저장 프로시져)	- Top SQL 문장 - Full Table Scan 문장 - 과도한 CPU 자원 사용 - 과도한 메모리 자원 사용 - 배치(Batch)성 SQL 문장	- SQL 문장의 Plan 정보 분석 - DBMS system event를 통한 Full Table Scan 분석 - Performance Dictionary View를 통한 자원(CPU, Memory, Disk I/O) 사용량 판단 - OLTP 업무 시간대에 실행되는 Batch성 문장의 자원 사용 경향을 분석하거나, 또는 전반적인 비즈니스 로직을 분석한다.
공유메모리 영역	- Buffer Cache 적중률(Hit%) - Shared Pool 적중률 (Library cache, Dictionary cache) - Cache 된 Object의 판단 - SQL TEXT 검사 - Block 경합(Contention) 통계 - 공유 메모리 사용 요약 정보 - SQL 문장의 Parsing - Redo Log Buffer의 Wait	- 메모리 적중률은 각 공유 메모리 영역별 Performance Dictionary View를 참조하여 Hit %로 환산하여 측정 및 분석한다. - 이 적중률에서는 현재 Cache 되어 있는 Object의 정도를 포함하여야 한다. - SQL 문장에 대한 모니터링 및 분석은 자원 사용량을 기준으로 자세하게 수집한다. - 메모리 블록의 경합에 관련된 모니터링 및 분석은 Segment의 Block 레벨까지 분석하여 경합이 발생하는 Object를 발견할 수 있어야 한다.
데이터베이스 파일 관리	- 데이터 파일의 분산배치 - 컨트롤 파일의 분산 및 이중화 - 리두로그 파일의 분산 및 이중화 - 아카이브 모드의 운용 - Tablespace의 여유 공간 검사 - datafile의 상태 검사 - Tablespace의 단편화 (Fragmentation)	- 데이터 파일, 컨트롤 파일, 리두로그 파일의 분산 배치 상태는 Performance Dictionary View를 분석하여 Disk I/O가 집중하여 발생하는 경향을 판단하여야 한다.(Disk I/O는 자료사전, 데이터, 인덱스, 정렬영역, 롤백영역 등에서 분산되어 이루어져야 한다.) - 아카이브 디렉토리는 별도로 마운트된 디스크 영역을 사용하는지 관찰한다. - 테이블스페이스의 단편화 경향은 스토리지 관련 Performance Dictionary View를 통하여 판단한다.

세그먼트 관리	– Table의 storage parameter – Index의 storage parameter – Extent 숫자 및 크기 – 세그먼트 단편화(Fragmentation) – 한 Block에 있는 Row의 통계정보 – 테이블에서 data를 보유한 block 수 – database 내에서 가장 size가 큰 segment의 검사 – table의 extent 수 검사	– 과도한 Extent의 할당, 잘못된 스토리지 파라미터, 단편화 등에 대한 정보는 Performance Dictionary View를 통하여 모니터링 / 분석한다. – 세그먼트의 블록, 로우, 단편화 등에 대한 정보는 통계정보를 수집한 후 각각의 세그먼트 관련 뷰를 통하여 분석한다.
정렬영역	– 과도한 정렬 작업 프로세스 – 과도한 정렬 작업 SQL 문장 – 메모리 정렬 대비 디스크 정렬 작업의 비율 – 임시 영역 세그먼트 할당 및 해제 – 정렬(Sort) 통계 정보	– 과도한 Extent의 할당, 잘못된 스토리지 파라미터, 단편화 등에 대한 정보는 Performance Dictionary View를 통하여 모니터링 / 분석한다. – 세그먼트의 블록, 로우, 단편화 등에 대한 정보는 통계정보를 수집한 후 각각의 세그먼트 관련 뷰를 통하여 분석한다.
롤백세그먼트 관리	– 롤백 세그먼트의 대기(Wait) – 블로킹(Blocking) 세션(Session) – 트랜잭션당 적절한 롤백 세그먼트 크기 조정/관리 – 롤백 세그먼트 내의 트랜잭션 정보 – 롤백 세그먼트의 저장영역 파라미터 – 롤백 세그먼트의 경합(Contention) Check 및 Stats Check	– 롤백 세그먼트의 대기(Wait) 정보 모니터링은 Performance Dictionary View의 롤백 상태 정보 뷰를 이용한다. – 이 뷰를 통하여 Wait 세션의 대기 정보를 분석하고 이를 바탕으로 적절한 조정(Tuning)의 근거로 삼아야 한다.
Locking 관리	– Deadlock 정보 검사 – Database 내 모든 Lock 정보 Object 정보 포함 – Redo Log Buffer Latch Contention Check – 모든 Latch Contention 정보 – 세션이 소유하는 Lock 정보	– 데이터베이스의 Locking은 크게 두 가지로 나뉘어지는데, 자원에 대한 Locking과 Data에 대한 Locking로 구분되며, Performance Dictionary View를 통하여 모니터링 및 분석 가능하다. – 자원에 대한 Lock인 Latch는 그 관찰 및 분석을 통하여 성능 조정 및 용량 증설의 근거로 활용될 수 있다. – Data에 대한 Locking은 enqueue는 locking 관련 뷰를 통하여 정의하며, 특히 deadlock은 어플리케이션의 오류로 인하여 발생하므로 어플리케이션의 모니터링 및 분석과 병행되어야 한다.
유저(세션) 관리	– Process Idle Time Check – 가장 CPU를 많이 사용하고 있는 Process 정보 – Process, Session별 자원사용량 판단 – 특정 Cursor에서 실행 중인 SQL 정보 – 특정 Tablespace에 속한 User, 사용된 Space Quota 정보 – 현재 Transaction이 Active한 Session, Process, OS User 정보 – Database에 생성된 User 정보 – Process가 수행하고 있는 Transaction 정보 (SQL 정보) – Session이 수행하고 있는 Transaction 정보 및 Undo Size	– Process/Session의 상태를 모니터링 하기 위해서는 Performance Dictionary View의 session 관련 뷰를 참조하여 분석한다. – 이 뷰를 통하여 세션 정보(접속한 노드 및 프로그램, 데이터베이스 내부유저 등)를 모니터링 하여야 하며, 내부적으로 각 세션이 사용한 CPU Time, Disk I/O 량, Idle Time 등을 분석하여야 한다. – 각 세션의 세부 활동을 분석하기 위해서는 세션이 현재 수행하고 있는 SQL 문장을 검사하여 Plan을 분석하는 방법을 사용한다.

※성능 측정 환경 하에서의 성능 정보의 수집은 기 계획된 주기로 수집

나. 성능 개선 방안 수립

(1) 대상 선정

성능관리에서 성능 개선을 위한 조정(튜닝)의 대상으로 선정되는 항목들은 단순하게 임계치를 초과하는 기준에 의해서 선정될 수는 없다. 개선 대상이 되는 항목은 업무 목표를 달성할 수 없는 상태의 근본원인을 선정하여 그 자원의 효율성을 개선할 수 있는 것을 대상으로 삼아야 한다.

□ 선정 기준 및 고려사항

▶ SLA를 위반한 성능 분석 결과
- 서비스 시간
- 응답속도
- 배치(batch) 적시 처리율
- 시스템 장애 발생 건수
- 동일 장애 발생률
- 시스템 장애 조치 시간
- 서비스 요청 적기 처리율
- 1차 Call 처리율
- 2차 Call 처리율
- 변경요청 적기 처리율
- 변경 적용시 오류 건수
- 고객만족도

▶ 성능지표(임계치)를 초과하는 요소
- 각 측정항목별 지정된(수치화된) 임계치(Threshold)
- 사용자(End-User)가 요구한 임계치

- 업무 대상으로부터 환산된 임계치
- 제작사(Vendor)로부터 권고된 임계치
- 기타 운영상의 목표로부터 설정된 임계치

▶ 기준치(Baseline)와 비교하였을 때 현격한 성능 차이를 보이는 요소
- 과거 성능 테스트 자료와의 비교결과 급격한 변화가 있는 요소
- 외부 툴 또는 프로그램을 이용하여, 정상적으로 운영될 때의 기준치 데이터와 현재 상태의 데이터를 지속적으로 비교하도록 설정한 경우 현격한 분석 값의 격차를 보이는 측정 항목을 개선 대상으로 선정한다.

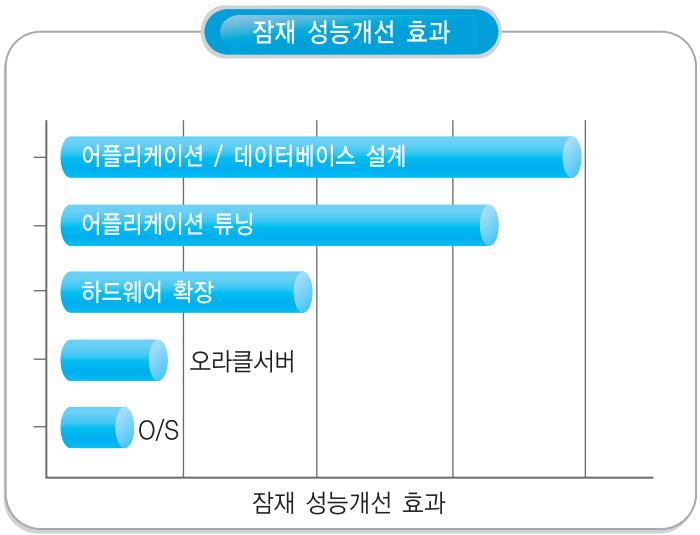
▶ 어플리케이션 담당자 및 사용자의 성능 조정 요청 내용
- 어플리케이션 담당자 또는 개발자로부터 특정 SQL 문장 또는 저장 프로시져(Stored Procedure)에 대한 성능 분석 및 개선의뢰가 있는 경우, SQL 튜닝을 통하여 처리하게 됨.
- 어플리케이션의 특정 성능(기능)을 수행하기 위하여 DBMS의 자원(CPU, 메모리, 디스크 I/O)의 할당, 파라미터의 조정을 요청한 경우

▶ 기타 성능 조정 요구사항이 발생한 대상 항목

(2) 방안 수립

일반적으로 DBMS의 성능개선 효과는 어플리케이션 로직 및 데이터베이스 스키마 설계 레벨에서 가장 큰 효과를 볼 수 있으나, 이것은 운영 상태에서 변경/조정하기는 매우 힘들다. 따라서 2차적으로 가장 큰 효과를 볼 수 있는 어플리케이션 튜닝(SQL문장 및 저장 프로시져)를 고려하는 것이 바람직하다. 하드웨어의 증설을 통한 용량 변경 성능조정이 그 다음으로 영향력이 크며, 오라클 서버의 파라미터 튜닝 및 데이터베이스 파일의 재배치, 그리고 운영체제 튜닝(Kernel Parameter Tuning) 등이 따른다.

성능 개선을 위한 방안 수립은 이 같은 개선 효과 및 영향도를 고려하여 수립되어야 한다. 가장 효과를 크게 미칠 수 있는 범위에서 출발하여 직접적으로 사용자(End-User)가 원하는 성능을 제공하는 포인트에 집중한다.



(그림 4-9) DBMS의 성능개선 효과

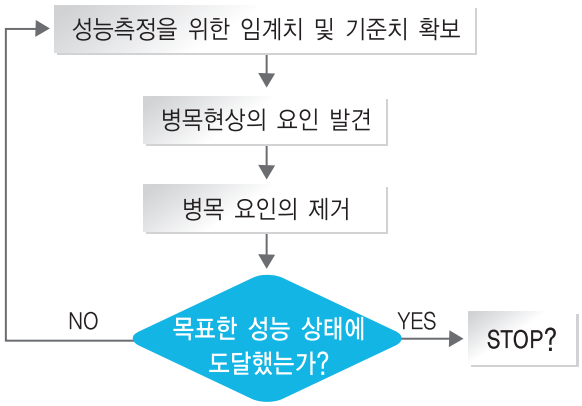
□ 방안 수립

성능 분석 결과, 개선이 필요한 경우 성능관리 책임자 및 관련 담당자(시스템 관리자, 데이터베이스 관리자, 응용 소프트웨어 관리자, 네트워크 관리자 등)와의 협의 하에 개선 대상 및 방안을 작성한다.

▶ 개선 방안에 포함되어야 할 내용
- 성능 개선이 업무 대상에 미치는 영향도 분석
- 성능 개선을 위한 참여자

- Trade Off : 한 항목의 성능을 개선하였을 경우 다른 항목의 성능이 나빠질 수 있으므로 반드시 Trade Off를 고려하여 성능 개선 방안을 수립하여야 한다(병목현상은 전이(轉移)되기 때문이다).
- 성능 개선의 목표 : 구체적으로 응답시간, 또는 처리량(트랜잭션 량)으로 환산한 값
- 연계분야와의 인터페이스

▶ 개선 프로세스



(그림 4-10) 개선 프로세스

개선 프로세스는 병목요인을 발견하고 이것을 제거해 나가는 과정을 반복하는 형태로 진행된다. 가장 중요한 것은 사용자가 만족할만한 성능 상태를 뚜렷한 수치로 환산하여 명확한 목표를 설정한 상태에서 프로세스를 진행하여야 한다는 것이다.

RDBMS 성능 개선 프로세스에서 조정(튜닝)의 종료는 명확한 목표에 대한 부분적인 종료가 있을 뿐, 전체 성능관리 프로세스 상에서 볼 때, '종료'란 있을 수 없다. 성능은 새로운 사용자의 추가, 어플리케이션의 수정/패치, 시스템의 하드웨어적인 변화, Job의 특성 등등 여러 가지 요인에 의하여 다시, 나빠질 가능성이 여전히 존재하기 때문이다. 따라서 성능 조정(튜닝) 프로세스는 지속적으로 진행되어야 한다.

다. 성능 개선 실행

성능 개선은 하나의 시스템을 구성하고 있는 다양한 계층별 요소들을 종합적인 시각에서 관찰하고 진단하여 현재 시스템의 상태를 평가하고 현재의 성능저하 요소 혹은 잠재되어 있는 미래의 성능저하 요소를 찾아내어 조정(튜닝)하고 최적화함으로써, 현재의 시스템 성능 뿐 아니라 향후의 시스템 성능을 보장하기 위한 일련의 작업이라고 할 수 있다. 데이터베이스 시스템에 대한 단계별, 계층별 성능조정(튜닝) 분야 및 세부 방법은 다음과 같다.

(1) 단계별 성능 조정(튜닝)

성능 조정이란 단지 시스템의 운영 및 사용 중에만 필요한 것이 아니다. 임의의 시스템을 분석, 설계, 개발 및 테스트, 운영하는 단계에서 모두 필요하다. 또한 데이터베이스를 근간으로 구축되는 시스템은 모든 단계에서 DBMS와 연관된 사항들을 고려하여 작업을 해야 하며, 각 단계별 성능 조정(튜닝)의 분야와 세부적인 내용들을 명시하여야 한다.

〈표 4-33〉 DBMS 구축 및 운영 단계별 성능관리

단계	튜닝포인트	체크포인트
계획/ 분석	- 업무프로세스 최적화 - 시스템 구조 - 용량산정	- 연계 업무간 데이터 흐름분석 - 요구 응답시간 기준 설정 - 트랜잭션의 처리량, 안정성, 유지보수성 고려
설계	- 데이터의 물리적 설계 - 어플리케이션 설계	- 설계 표준 설정 - 물리적 설계 전환시 성능관점 견지
개발/ 테스트	- 인덱스 정책 - SQL, PL/SQL 구사의 효율성 - 옵티마이저의 이해 및 적용도	- 개발 표준 설정 - 개발자 교육 - 운영환경에 가까운 개발환경 구축 (충분한 데이터양 확보) - 개발 장비 튜닝
운영	- 운영체제의 튜닝 - 네트워크의 튜닝 - 데이터베이스 튜닝 - 어플리케이션의 튜닝	- 응답시간 관점에서의 성능 평가 수행 - 계층 요소별 종합적 시각 - 지속적인 모니터링 및 튜닝

(가) 계획/분석 단계

계획/분석 단계에서는 전체적인 성능 및 안정성을 고려하여 분석을 해야 하는데, 이때 중요한 것은 업무 프로세스의 최적화, 시스템 구조(Technical Architecture) 설정과 용량 산정(Capacity)이다. 업무 프로세스의 최적화는 기간 시스템을 전산화 하면서 비효율적인 프로세스를 개선함으로써 전체적인 성능을 향상시킬 수 있다. 시스템의 구조는 트랜잭션의 처리량, 안정성, 유지보수성 등을 고려한 구조로 결정해야 하며, 용량 산정은 구축하고자 하는 업무의 트랜잭션 및 동시 사용자 수, 데이터의 증가치 등을 분석한 예상치 들을 적용하여 산정한다. 여기서 주지해야 하는 것은 정확한 용량을 산정하려고 노력하기 보다는 대략적인 경험치를 이용하여 산정한다는 것이다. 대략적인 경험은 과거의 BMT(Bench Mark Testing)의 결과자료나 비슷한 시스템을 오픈할 때 테스트한 자료 등이 좋은 참고 자료가 될 수 있다. 또한 용량 산정 시에는 약간 여유 있게 산정하는 것이 바람직하다. 향후 시스템 오픈 후에 용량이 부족한 것 보다는 여유 있는 것이 결과적으로 잘 예측된 용량 산정이 되기 때문이다. 이 단계에서 각 업무별 요구 응답시간 기준을 수립하는 작업이 수행되어야 한다. 용량 산정은 구축하고자 하는 업무의 트랜잭션 및 동시 사용자 수, 데이터의 증가치 등을 분석한 예상치 들을 적용하여 산정한다. 용량산정 시에는 실제 예상되는 용량보다 30% 정도의 여유를 두고 산정하는 것이 일반적이다.

(나) 설계 단계

설계 단계에서는 논리적 설계 보다는 데이터의 물리적 설계 시에 성능에 관련된 작업이 함께 고려되어야 한다. 테이블의 정규화 또는 역정규화를 통하여 성능 조정을 할 때에는 매우 큰 비용이 동반되므로, 논리 설계 시에 도출된 ERD(Entity Relationship Diagram)를 기초로 하여 구축하고자 하는 시스템의 구조와 성능을 최대한 고려하여 물리적 설계를 하여야 한다. 실제로 설계 및 구축이 완료된 후, 테이블 구조를 변경하게 되면 어플리케이션의 수정이 불가피해지게 된다. 이에 따라, 관련된 오브젝트들 역시 수정하게 되므로 조정(튜닝) 비용이 상승

하게 되는 것이다. 따라서, 설계 단계에서 요구 응답시간, 분산 데이터베이스 환경, 동시사용자의 수, 데이터의 크기, 병렬 프로세싱 등을 고려하여 철저한 설계 단계를 거쳐야만 최적의 DB 구축이 가능해 진다. 물론 어플리케이션 설계 역시 데이터베이스와 밀접하게 연동되므로 최적의 성능을 발휘할 수 있도록 어플리케이션을 설계하여야 한다.

(다) 개발/테스트 단계

개발 및 테스트 단계에서는 개발 표준을 설정하여 주지시키고, SQL, PL/SQL의 효율적인 구사방법, 옵티마이저의 특성 등에 대해 개발자들을 충분히 교육시키는 것이 매우 중요하다. 특히, 개발자들이 DBMS 옵티마이저에 대해 충분히 이해할 때 단순히 원하는 결과를 추출하는 수준의 SQL이 아니라 내부 처리량을 최소화 할 수 있는 SQL을 구사할 수 있게 되므로, 전체 시스템의 성능에 좋은 영향을 주게 된다. 모든 개발자들은 효율적인 SQL 구사를 위한 표준안, 또는 권고안을 따라서 프로그램 코딩을 하여야 하며, 프로젝트 관리자는 전체적인 RDBMS 시스템의 성능을 고려하여 개발자들이 작성한 SQL 문장에 대하여 개별적인 검토를 하여야 한다.

또한, 개발 및 테스트 환경을 가능한 실제 운영환경에 가깝게 구축하고 데이터량과 파라미터 설정을 운영환경과 동일하게 구성하여야 실제 시스템 오픈 후 실패를 겪을 확률을 줄일 수 있다. 개발 및 테스트 단계에서의 튜닝은 시스템 오픈의 성공을 좌지우지할 정도로 중요하므로 오픈 전에 튜닝을 많이 하면 할수록 안정된 오픈을 할 수 있는 밑받침이 된다.

(라) 운영 단계

마지막으로, 운영 단계에서 수행할 수 있는 튜닝 작업으로 어플리케이션 튜닝, 데이터베이스 튜닝, 운영체제 튜닝, 네트워크 튜닝이 있다. 개발 및 테스트 단계에서의 튜닝은 시스템 오픈의 성공을 좌지우지할 정도로 중요하다. 오픈 전에 튜닝을 많이 하면 할수록 안정된 오픈을 할 수 있는 밑받침이 된다. 많은 고객들이 시스템을 오픈할 때 어플리케이션의 완성도 및 구

조상의 문제, 성능의 문제 등으로 오픈을 연기하는 경우가 종종 있는데, 이는 분석, 설계 단계가 완벽하게 이루어 지지 않았기 때문이다.

실제로 대부분의 데이터베이스 시스템에서의 성능관리는 어느 정도 서비스 운영이 진행된 상태에서 시스템의 자원 사용 추이를 고려하여 진행하는 것이 일반적이다. 물론 운용 전에 완벽한 계획 하에 성능 저하 문제가 전혀 발생하지 않도록 관리하는 것이 최고의 프로젝트 관리이겠지만, 현실적으로 불가능에 가깝고, 운영 시 자원 사용 추이나 서비스 어플리케이션의 RDBMS 이용 경향을 판단하여 튜닝하는 것이 최적의 성능 조정 방법이다. 따라서, 운영 단계에서의 성능 조정은 성능 저하 지점(병목지점)에 대한 지속적인 모니터링과 튜닝, 피드백, 그리고 모니터링을 다시 반복하게 되는 것이다.

(마) SLA 관점에서의 성능조정

성능관리 및 튜닝을 수행함에 있어 가장 우선적으로 이루어져야 할 것은 과연 얼마만큼의 성능을 만족스러운 수준이라고 인정할 수 있는가? 즉, SLA(Service Level Agreement)를 정의하는 것이다. 너무도 당연한 말이겠지만, 우리가 전산시스템을 운영하는 목적도 사용자 곧 고객을 위한 것이며, 결국 모든 작업은 사용자로부터 시작되어 사용자에게서 끝나게 되어 있다. 또한, 한 사람이 건강한 상태인지 아닌지는 당사자가 가장 잘 알듯이 우리의 시스템의 성능이 좋은지 나쁜지는 사용자가 가장 먼저 느끼게 되어 있다. CPU 사용률이 정상 수준을 유지하고 버퍼 캐쉬, 메모리 히트율이 아무리 좋아도 사용자가 체감하는 성능이 느리면 우리의 시스템 어딘가에 문제가 있는 것이다. 만약, 실제로 성능저하가 아니라, 심리적인 원인에 기인하는 것이라고 판단될 수도 있겠으나, 그러한 경우라 하더라도 SLA 수준을 재정의하여 사용자의 기대치와 우선 순위를 조정해야 하는 것 역시 성능관리자의 몫이다.

사용자가 경험하는 응답시간에는 데이터베이스에서의 SQL 처리시간, 어플리케이션 서버에서의 로직 처리시간, 네트워크 경유시간, 클라이언트에서의 화면처리 시간이 모두 포함된다. 일반적인 경우에 실질적으로 응답시간의 대부분은 데이터베이스 처리과정에서 소요된다. 데

이터베이스에서의 처리시간도 좀 더 세분화시켜 살펴볼 수 있는데 실제 연산 및 블록 액세스에 소모되는 CPU 서비스 시간뿐만 아니라 CPU나 메모리 자원을 할당 받기 위한 대기시간, 네트워크 교신이나 디스크 I/O를 위한 대기시간, Enqueue나 Latch와 같은 데이터베이스의 제어자원에 대한 대기시간 등이 그것이다.

이처럼 성능분석을 위해서는 사용자의 최종 응답시간의 관점에서 평가를 수행해야 하며 응답시간을 구성하는 각 계층 요소들에 대한 전문적인 지식과 더불어 종합적인 시각이 요구된다.

(2) 용량 증설

H/W 용량산정은 공공기관의 ISP사업 중 기술 아키텍처의 설계나 소요예산에 대한 추정 단계에서 적정한 시스템 규모를 파악하기 위해서 실시하거나 공공기관에서의 제안시점, 그리고 정보시스템 구축에서의 아키텍처 분석 및 설계 시점, 또는 운용 중인 시스템의 증설 및 확장을 위하여 산정이 이루어진다. 특히 국내 공공부문 정보화사업에 있어서는 H/W 용량산정은 정보화사업을 제안하는 시점에 주로 이루어지고 있으며, 시스템 운용 단계에서 용량산정을 별도로 실시하지 않는 것이 현실이다.

실제로, 용량산정 형태는 크게 두 가지로 구분할 수 있다. 우선 기존시스템에 대한 증설 혹은 교체를 위한 시스템 용량산정 방식이다. 이는 기존 운영 중인 시스템을 바탕으로 업무의 일부 확장 혹은 기존업무의 재개발 등 운영 중인 시스템을 바탕으로 용량산정을 하는 경우이다. 다음으로는 신규시스템에 대한 용량산정으로 이전에 없던 새로운 업무를 위한 시스템 구축에 따른 도입 시스템의 용량을 산정하는 경우이다. 이 2가지 경우에 있어서 운영시스템의 증설 혹은 교체를 위한 용량산정은 신규시스템 구축에 따른 용량산정에 비하여 복잡하지 않다. 예를 들면 OLTP 데이터베이스 시스템의 CPU 용량산정 하는 경우에 있어서 용량산정을 위한 항목 중 가장 기본이 되는 요소인 동시사용자 수나 트랜잭션 수의 추정은 신규시스템 보다는 재구축 혹은 용량 증설 및 확장 시스템이 훨씬 용이하기 때문이다.

성능관리 분석 대상 중 용량 증설을 통하여 비즈니스 목표에서 원하는 성능을 제공하기 위해서는 다음과 같은 정보들이 용량관리 분야로 전달되어야 한다.

(가) RDBMS 시스템을 위한 CPU 용량산정 시 고려 요소

- 동시사용자수
- 트랜잭션 처리수
- 기본 TPMC 보정
- Peak Time 보정
- 데이터베이스 크기 보정
- 어플리케이션 복잡도 보정
- 사용자 복잡성 보정
- 어플리케이션 구조 보정
- 어플리케이션 부하 보정
- 네트워크 보정
- 클러스터 보정
- 여유율 보정

※ 각각의 용량 산정 시 고려 항목에 대해서는 '정보시스템 규모별 용량산정 기준 연구(2004, 한국전산원)'를 참조한다 (문서번호 : NCA IV-RER-04019/2004.10).

정보시스템에 대한 용량산정에 있어서의 주요한 논란은 CPU의 용량 산정에 대한 부분인데, 성능관리 분야에서는 이러한 용량산정 주요 대상인 CPU를 중심으로 성능에 영향을 미치는 요소를 파악하고 이를 용량산정을 위한 세부 항목으로 반영하여야 한다. 일반적으로 정보시스템의 성능에 영향을 미치는 요소는 시스템 설계 시 CPU 성능에 직접적인 영향을 미치는 요소와 응용을 전제로 한 성능 요소로 구분할 수 있다.

(나) RDBMS 시스템을 위한 메모리 용량산정 시 고려 사항

메모리에 대한 용량산정은 시스템 영역, 데이터베이스 사용 영역, DB접속 프로세스 영역, 어플리케이션 영역 등 OS 및 DBMS, 미들웨어 등의 운영에 따른 기존 필요메모리에 Peak Delay 보정, 피크타임 보정 튜닝을 위한 보정등의 보정치와 시스템의 여유율을 반영하고 있

다. 특히 이들 산정항목 중 데이터베이스 영역과 DB접속프로세스 영역, 어플리케이션 영역 등은 동시 사용자수에 일정한 메모리를 할당하는 방식으로 계산한다.

(다) 용량증설시의 하드웨어 산정식

용량증설을 통해 성능 향상이 필요한 경우에는 용량증설이 선 수행되어질 수 있도록, 개선 방안을 수립되어야 한다.

성능관리 분야로부터 요구되어지는 용량 증설은 객관적으로 엄밀히 분석된 성능관리 수집 데이터로부터 출발한다. 성능 조정을 거치고도 개선 효과가 없을 것으로 판단되면, 용량 증설을 위한 Capacity Planing 및 Sizing을 위하여 다음의 기초 데이터를 용량관리 분야로 이관하여야 한다. 용량산정을 위한 DBMS 서버의 CPU, Memory, Disk 용량 산정식은 〈표 4-34〉를 참조한다(일반적으로 DBMS 서버의 CPU는 tpmc(분당 처리할 수 있는 트랜잭션의 수)를 기준으로 용량이 산정되며, 메모리 및 디스크 용량은 어플리케이션의 사용량을 기준으로 계산된다).

〈표 4-34〉 DBMS 서버를 위한 CPU/메모리/디스크 용량 산정 공식

구 분	산정수식
CPU	CPU 용량(tpmC) = 동시사용자 수×트랜잭션 수×기본TPC보정치 ×Peak Time 보정치×CPU부하 보정치×응용 프로그램 복잡도 보정치×네트워크 보정치 ×클러스터 보정치×여유율 보정치
메모리	메모리 용량(MB) = {OS 및 기본영역+프로세스 수+응용 프로그램 보정치}×버퍼 캐쉬 보정치×클러스터 보정치 ×여유율 보정치 +{데이터베이스 공유 메모리}
디스크	내장디스크 용량(MB) = {시스템OS영역+응용프로그램 영역 +상용 소프트웨어 영역}×SWAP 영역 ×여유율 보정치 외장디스크 용량(MB) = {DB영역+백업영역}×RAID 영역 ×여유율 보정치

4 2.4 응용 소프트웨어

가. 데이터 수집 및 분석

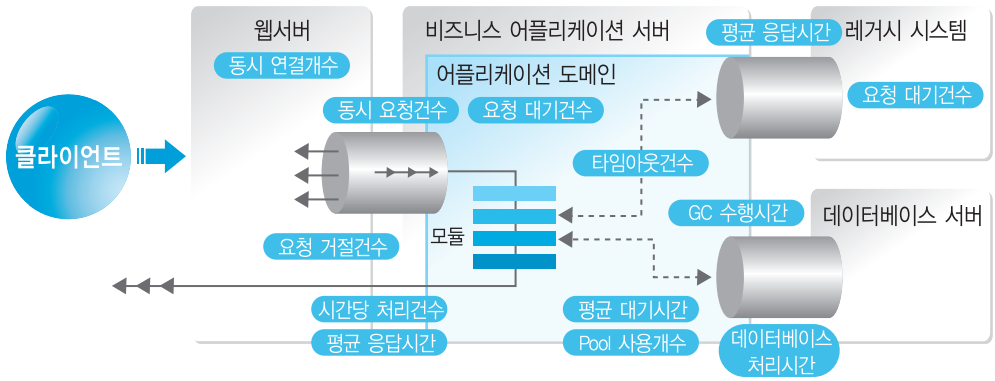
비즈니스 어플리케이션이 정의된 성능 목표를 만족하는지, 성능을 저하시키는 주요 병목 현상은 무엇인지, 성능 저하에 대한 최적의 개선방안은 무엇인지 등을 판단하기 위해서는 비즈니스 어플리케이션을 포함한 전체 시스템 구성요소(서버, 네트워크, 데이터베이스 등)에 대한 성능을 측정하고 적절한 지표 값들을 수집하여 이를 종합적으로 분석하는 것이 필요하다. 이때, 사용되는 주요 성능 지표에는 응답 시간, 시간당 처리량 및 자원 사용량(어플리케이션 수행 중 사용되는 CPU, 메모리, 디스크 I/O, 네트워크 대역폭 등) 등이 있다.

어떤 성능 지표를 사용할 것인지, 각 성능 지표의 목표 값을 얼마로 정의할 것인지가 효과적인 성능 측정 또는 성능 관리를 위해서 가장 중요한 부분이라고 할 수 있는데, 모든 정보시스템에 동일하게 적용할 수 있는 절대적인 성능 지표가 존재하는 것이 아니므로 각 기관의 비즈니스 목표와 도입된 시스템의 특성들을 고려하여 해당 시스템에 적합한 성능 지표 및 목표를 결정하여야 한다(3.1절, 성능관리의 개념 참조).

일반적으로는 대량의 트랜잭션을 동시에 처리하여야 하는 백엔드 (back end) OLTP 시스템의 경우에 시간당 트랜잭션 처리량이 개별 트랜잭션의 응답 시간보다는 중요한 지표로 사용되며 내부 사용자 시스템 또는 프런트엔드(front end) 시스템의 경우에는 동시 사용자 수를 고려한 개별 트랜잭션의 응답 시간을 중요한 성능 지표로 사용하고 있다.

(1) 측정 항목

(그림 4-11)은 웹 어플리케이션이 포함된 일반적인 형태의 비즈니스 어플리케이션에 대한 성능 측정 지표와 각 지표들이 서비스 요청처리 단계별로 어떤 관계가 있는지를 나타내는 예이다.



(그림 4-11) 주요 성능 측정 항목

본 절에서는 (그림 4-11)에서 나타난 측정 항목들을 중심으로 3.2.4절 응용 소프트웨어 성능 관리 대상 및 범위의 분류 기준에 따라 응용 소프트웨어의 성능 측정 항목을 응용 프로그램, 응용 플랫폼, 응용 솔루션으로 나누어 설명하도록 한다. 여기에서 기술되는 각 측정 항목들은 일반적인 기준일 뿐이므로 구체적인 측정 항목 및 측정 주기는 각 기관에 도입된 시스템의 특성을 고려하여 가감하여 적용되어야 한다.

각 측정 항목 중 응용 프로그램, 응용 플랫폼, 응용 솔루션에 모두 포함된 응답 시간/트랜잭션 처리량은 반드시 부하량을 기준으로 측정하여야 한다. 이는 해당 지표의 목표 값 또한 부하량을 기준으로 정의되어야 함을 의미한다. 일반적으로 부하량은 사용자, 데이터 및 비즈니스 로직의 세 가지 측면에서 정의할 수 있다.

사용자 측면의 부하량은 동시에 서비스를 요청하는 사용자의 수를 의미하는 것으로 동시라는 개념을 어떻게 정의하느냐(Think Time 고려 여부)에 따라 동시 사용자(또는 트랜잭션) 또는 동시 요청자(본 지침서에서는 특정 시점에 서버에서 처리 중인 클라이언트 요청 건수로 정의함)로 표현(3.1.2절 성능 지표 참조)된다.

데이터 측면의 부하량은 개별 사용자의 요청을 처리하기 위하여 접근해야 하는 데이터의 크기와 건수를 의미하며 비즈니스 로직 측면의 부하량은 조건에 따른 개별 사용자의 요청을 처리하기 위하여 수행해야 하는 비즈니스 로직의 복잡도를 의미한다. 간단한 예로, 1건의 주문 처리를 위해 3번의 SQL을 사용해야 하는 것과 30번의 SQL을 사용해야 하는 것의 차이를 나타낸다.

(가) 응용 프로그램

〈표 4-35〉 응용 프로그램 측정 항목

대 상	측정항목	측정주기
응답 시간/ 트랜잭션 처리량	– 부하량에 따른 응용 프로그램의 평균, 최소, 최대, 90% 응답 시간 및 분포도와 대기 시간 – 부하량에 따른 응용 프로그램의 초당 트랜잭션 처리 건수, 처리 시간 및 처리량 추이 – 타임아웃 발생 건수, 타임아웃 발생 시의 응답 시간	실시간
메모리 사용	– 응용 프로그램 코드 및 라이브러리의 메모리 크기 – 시간당 또는 일별 메모리 증가량 및 증가율	정기
데이타베이스 처리	– 사용된 SQL의 처리 시간(악성 트랜잭션 확인)	실시간
오류 및 예외	– 오류 및 예외 발생 여부, 유형 및 패턴	실시간
배치 실행 환경	– 배치 프로그램 수행 시간, 선 · 후행 작업 결과 및 자원 사용량	실시간

응용 프로그램 측면에서 타임아웃을 적용하는 대표적인 경우는 원격 노드의 응용 프로그램, 응용 플랫폼 또는 응용 솔루션 및 외부 기관과의 시스템 연계 시 TCP/IP 소켓 등을 사용하여 데이터를 송수신하고자 하는 경우이다.

타임아웃이 설정 되어 있지 않거나 지나치게 큰 값으로 설정된 경우에는, 원격 시스템의 응답 시간이 급격히 저하되거나 문제가 발생하여 응답을 받지 못하는 경우 다른 작업을 처리하지 못하고 계속하여 응답을 기다리게 되므로 시스템의 전반적인 응답시간 저하 및 트랜잭션 처리량 저하로 이어지게 된다. 반대로 타임아웃 시간을 지나치게 짧게 설정한 경우에도 요청 처리에 사용된 시스템의 자원(비용)을 헛되이 낭비하게 될 뿐만 아니라 전체적으로 트랜잭션 처리량을 저하시키게 되므로 비즈니스 어플리케이션의 목표 및 성능 측정 결과를 바탕으로 적절한 타임아웃 시간을 설정해야 한다.

배치 프로그램들은 일반적으로 온라인 프로그램의 사용량이 적은 야간 또는 새벽 시간대에 주로 수행되는데 정의된 처리 시간 내에 정상적으로 수행되는지 확인하여야 한다. 대개는 해당

시스템을 운영하면서 처리해야 할 데이터의 증가로 인해 처리 시간이 늘어나게 되는 것이 일반적이다. 이럴 경우, 배치 프로그램에 대한 성능 조정(튜닝) 또는 수행 계획을 재조정하는 것이 필요하다.

만일, 온라인 업무 처리의 수행을 위하여 빈번하게 배치 프로그램이 수행되는 경우에는 해당 프로그램의 자원 사용량, 처리시간 및 온라인 프로그램 수행에 대한 성능상의 영향(경합 또는 대기 시간)을 측정하는 것이 필요하다. 또한, 배치 프로그램 수행으로 인한 온라인 프로그램과의 데드락(Dead Lock)이 발생하지 않는 지 확인하여야 한다.

(나) 응용 플랫폼

〈표 4-36〉 응용 플랫폼 측정 항목

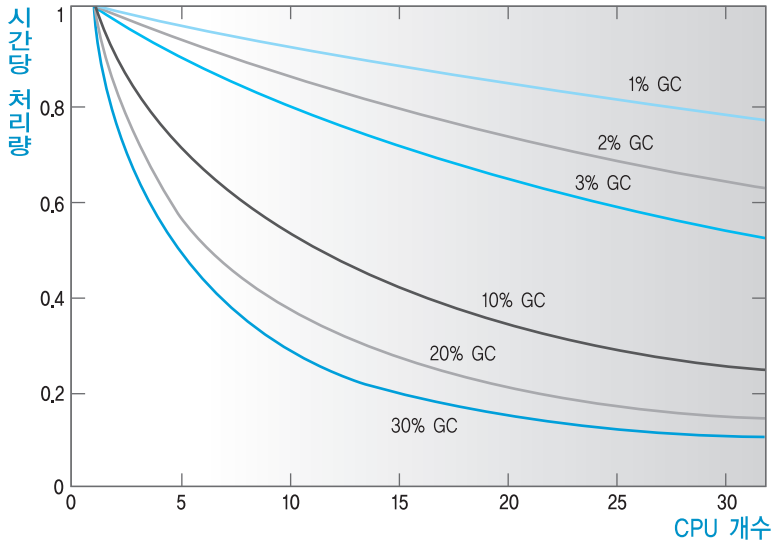
대 상	측정항목	측정주기
응답 시간/ 트랜잭션 처리량	- 부하량에 따른 평균, 최소, 최대, 90% 서비스 시간 및 분포도와 대기 시간 - 부하량에 따른 초당 트랜잭션 처리(요청) 건수, 전체 요청(처리) 건수 및 처리량 변화 - 서비스(커넥션) 타임아웃 발생 건수, 타임아웃 발생 시의 응답 시간	실시간
대기큐/ 대기 시간	- 대기 큐에 저장된 요청들의 실시간, 평균, 최대 요청 개수 및 변화 추이 - 대기 큐에서 소요된 평균, 최대 대기 시간 및 대기 시간 변화 추이 - 대기 큐에 저장되지 못하고 거절된(rejected) 요청들의 개수 및 변화 추이	실시간
프로세스 (쓰레드) 상태 및 개수	- 프로세스 또는 쓰레드의 실시간 서비스 상태 및 자원 (CPU, 메모리, 디스크 I/O 등) 사용량, 메모리 사용량 추이 - 시간당 GC(Garbage Collection) 횟수 및 처리 시간, GC 전, 후의 메모리 사용량 추이 (J2EE, .NET) - 어플리케이션 프로세스 또는 쓰레드의 평균, 최대 실행 개수	실시간
세션 상태 및 개수	- 클라이언트 세션의 메모리 사용량 - 클라이언트 세션의 평균, 최대 개수 및 변화 추이	실시간 및 정기
통신 큐, 채널 상태	- 클라이언트 연결 요청에 대한 시간당 drop 건수 - TCP, UDP 등의 시간당 오버플로우(over flow) 건수 및 비정상 TCP 연결 (SYN_SENT, FIN_WAIT_2, CLOSE_WAIT) 건수 - 사용하고 있는 파일 디스크립터(descriptor) 개수	실시간 및 정기

자원 (Resource) Pool	– 데이터베이스 커넥션 풀(Connection Pool), 쓰레드 또는 객체 풀 등의 실시간 및 평균, 최대 사용 개수	실시간 및 정기
오류 및 예외	– 응용 플랫폼의 오류 및 예외 발생 건수, 유형 및 패턴	실시간
부하 분산	– 클라이언트 요청에 대한 실시간 부하 분산 상태	실시간

대기 큐에 사용자의 요청이 쌓이게 되는 대표적인 원인은 두 가지로 분류할 수 있는데 하나는 사용량의 증가이고 나머지는 서비스 처리에 문제가 발생하여 처리 시간이 급격히 증가하였기 때문이다. 사용량이 증가한 경우는 시스템 측면의 장애 또는 문제로 보기 어려우나 이럴 경우 다른 서비스 또는 전반적인 응답 시간의 심각한 저하로 이어지지 않도록 하는 방안이 마련되어야 한다. 일반적으로 대기 큐에 쌓이는 요청의 개수가 증가하게 되면 동일한 부하량 이상에서는 대기 큐에 쌓이는 요청의 개수가 급격히 지속적으로 증가하게 된다. 따라서, 대기 큐는 최대 허용 개수를 적절히 설정하는 것이 필요하며 이를 초과하는 요청에 대해서는 거절(reject) 하도록 하는 방법이 일반적으로 적용된다. 서비스 처리에 문제가 발생한 경우에는 평균 서비스 처리 시간이 증가하는 추세를 보이게 되므로 이럴 경우 데이터베이스 시스템, 관련 연계 시스템 또는 해당 노드의 시스템 환경의 문제가 발생하였는지 여부 및 구간별 응답 시간을 확인하도록 한다.

J2EE 또는 .NET을 기반으로 하는 비즈니스 어플리케이션의 경우 사용되지 않는 메모리를 시스템 내부적으로 정리하는 GC(Garbage Collection)라는 작업을 수행하게 되는데 이의 수행 횟수 및 처리시간은 시스템의 자원(CPU, 메모리) 사용량에 비례하여 증가하는 것이 일반적이다. 시스템의 규모 및 GC 적용 방식에 따라서 GC 수행으로 인한 비즈니스 어플리케이션의 응답 시간 및 시간당 처리량이 심각하게 저하될 수도 있으므로, 비즈니스 어플리케이션의 특성을 고려하여 조금씩 자주 GC를 수행할 것인지 아니면 한번에 많은 양의 GC를 수행할 것인지를 결정해야 한다.

(그림 4-12)는 일반적인 GC의 수행시간과 시스템 확장성과의 관계를 나타낸 그래프이다.



(그림 4-12) GC와 시스템의 확장성

(그림 4-12)에서 알 수 있는 중요한 사실은 개발(일반적으로 소규모 시스템에서 수행) 당시에는 간과되었던 GC의 성능 지표들이 운영 시스템에서는 심각한 성능 저하로 이어질 수 있으며, 운영 시스템의 성능 향상에 GC의 성능 조정(튜닝)이 상당한 기여를 할 수도 있다는 것이다.

예를 들어, 비즈니스 어플리케이션 수행 시간의 3%가 GC에 사용된다고 하면 CPU 4개 규모의 개발 서버에서는 GC로 인한 시간당 처리량 손실이 5%에도 못미치지만 CPU 30개 규모의 운영 시스템에서는 45%의 손실이 발생하게 된다. 이 때에는 GC 수행시간이 시스템의 성능에 있어 중요한 변수가 되므로 전체 수행시간을 줄이기 위해 힙(heap) 사이즈를 조정하거나 적절한 GC 컬렉터로 변경하게 된다.

메모리의 사용과 관련하여 또 다른 중요한 성능 측정 항목 중의 하나는 전체 및 동시에 사용중인 클라이언트 세션의 개수 및 크기이다. 이를 조절하는 가장 중요한 설정 파라미터 중의 하나는 세션 타임아웃으로, 클라이언트 세션 정보는 사용자가 웹 어플리케이션을 사용하고 있지 않아도 일정 시간(타임아웃 또는 어플리케이션 플랫폼별 자체 제거 시간) 동안 메모리에 유지되기 때문에 이 시간이 지나치게 길게 설정된 경우 사용되지 않는 세션 정보에 의한 메모리의 낭비가 많아지게 된다. 반대로, 타임아웃이 너무 짧게 설정된 경우에는 빈번하게 세션을 재 생성해야 하므

로 응답 시간 저하 및 추가적인 CPU 및 디스크 자원의 사용을 필요로 하게 된다. 따라서, 세션과 관련한 메모리의 사용을 최소화 하면서 시간에 따른 메모리 사용량이 거의 일정하게 유지될 수 있는 최적의 설정 값을 도출하는 것이 필요하다.

일반적인 비즈니스 클라이언트의 경우 서버에 요청을 보내기 전에 먼저 연결(TCP)을 맺어야 하는데 특히, 대규모 사용자를 대상으로 하는 웹 시스템의 경우에는 클라이언트의 연결 요청이 순간적으로 동시에 많이 발생할 수 있기 때문에 경우에 따라서는 연결 요청 자체가 폐기(drop)될 수도 있다. 클라이언트의 연결 요청은 연결 요청 큐(운영 체계에 따라 차이가 있음)에 저장되는 데 이 큐의 크기는 운영 체계의 파라미터(일반적으로 TCP 파라미터)와 어플리케이션 플랫폼의 설정 값에 의해 좌우되므로 연결 요청에 대한 폐기가 허용 수준 이상으로 발생되면 연결 요청 큐 및 관련 어플리케이션 플랫폼 파라미터를 변경할 필요가 있다.

연결 요청이 성공적으로 처리된 이후에는, 클라이언트는 사용자의 요청을 해당 서버로 전송하게 되고 서버에서는 이 요청을 처리한 후 처리 결과를 클라이언트로 보내게 된다. 어플리케이션 플랫폼에 따라 차이가 있으나 웹 시스템의 경우에는 일정 시간이 경과하면 연결을 해제하게 되는데 연결 상태가 장시간(보통 수십 초) FIN_WAIT_2 또는 CLOSE_WAIT로 지속될 경우에는 성능상의 문제가 발생할 가능성이 높다. 이러한 경우에는 응용 어플리케이션 내부의 문제(자원 미해제, 데드락 또는 무한 루핑 등)에 의한 것이거나 관련 플랫폼의 버그에 의한 것일 수 있다.

클라이언트의 요청을 서버에서 처리하는 과정에는 다양한 자원 Pool이 관련되어 있다. 이러한 자원 Pool은 초기화 비용(시간 또는 CPU 사이클)이 많이 소모되는 자원들(주로, 데이터베이스 연결, 비즈니스 어플리케이션 프로세스/객체/쓰레드 등)에 대하여 구성하는 것이 일반적인데 사이즈가 지나치게 크게 설정될 경우에는 메모리를 낭비하게 되는 요인이 될 뿐만 아니라 운영 체제 및 플랫폼의 관리 작업에 따른 오버헤드를 가중시켜 성능의 저하를 가져올 수 있다. 반대로, 너무 작게 설정된 경우에는 해당 자원에 대한 과다 경합이 발생하게 되어 대기 시간 증가로 인한 응답 시간이 증가하게 되고 경우에 따라서는 클라이언트의 요청 자체가 거절(reject)되거나 실패로 이어지게 된다.

클라이언트 요청 처리 도중 발생되는 어플리케이션 또는 플랫폼의 오류 및 예외는 이를 적절하게 처리하기 위한 다양한 추가 자원(CPU, 메모리, 디스크 I/O 등)을 요구하게 된다. 발생 빈도

가 높을 경우, 해당 시스템의 성능에 상당한 악영향을 미치게 되므로, 어플리케이션 로그 또는 플랫폼 로그를 정기적으로 분석하여 해당 오류 및 예외를 주기적으로 제거하는 노력이 필요하다.

(다) 응용 솔루션

〈표 4-37〉 응용 솔루션 측정 항목

대 상	측정항목	측정주기
구간별 수행 시간	- 평균 응답 시간, 대기 시간, 로드 시간, 데이터베이스 처리 시간	실시간 및 정기
대기 큐	- 대기 큐에 존재하는 평균 요청 개수, 대기 시간	실시간
메모리/버퍼	- 메모리 영역/버퍼 사용량, Hit율 또는 Miss율	실시간 및 정기
오류 및 예외	- 응용 솔루션의 오류 및 예외 발생 건수, 유형 및 패턴	실시간

3.2.4절, 응용 소프트웨어 성능관리 대상 및 범위를 기준으로 응용 솔루션에는 그룹웨어, CRM, ERP, SCM, B2Bi, BPM, ETL, EII, 디렉토리, 인증/권한관리 등의 도구들이 포함된다. 이들 응용 소프트웨어 도구들은 조직의 핵심 업무를 지원하는 것으로부터 데이터, 프로세스의 통합 및 응용 보안을 위한 것까지 그 성격이 다양하다. 따라서, 위의 표에서 제시된 일반적인 측정 항목 이외에 각 기관에 도입된 솔루션의 특성 및 적용 범위, 성능 목표 등에 따라서 적절한 측정 지표와 목표를 수립하고 이에 따라 성능 측정 및 조정(튜닝)을 수행하여야 한다.

(2) 측정 방법

(가) 측정 도구 및 고려사항

응용 소프트웨어(응용 어플리케이션, 응용 플랫폼, 응용 솔루션)에 대한 성능 측정은 정보시스템의 개발 및 운영 단계에서 다양한 도구 및 기법들을 활용하여 이루어지는데, 일반적으로 다음과 같은 것들이 포함된다 〈표 4-38〉.

〈표 4-38〉 응용 소프트웨어 성능측정 도구 및 기법

구분	설명	도구 예제	용도
시스템 모니터링 도구	운영 체제에 기본적으로 포함되어 있거나 추가가 가능한 도구로 시스템 자원 사용량 및 관련 이벤트에 대한 실시간 및 통계 정보를 제공	top, glance, topas, sar, vmstat, iostat, mpstat, prstat, lockstat, trapstat, netstat, nfsstat, system monitor(윈도우) 등	CPU, 메모리, 디스크 I/O 및 프로세스, 쓰레드 등의 자원 사용량 수집
네트워크 모니터링 도구	네트워크 패킷 또는 트래픽 부하에 대한 실시간 및 통계 정보를 제공	Ntop, sniffer, ethereal, snoop, iptrace, MRTG 등	네트워크 패킷 및 트래픽 부하량 수집 및 분석
시스템 관리도구	시스템의 운영 상태를 종합적으로 모니터링하여 실시간 및 통계 정보를 제공하고 수집 정보와 정의된 임계치에 따라 자동 통보 등의 기능을 제공	OpenView, Patrol, Tivoli, uniCenter, NetIQ 등	시스템의 전반적 운영 상태 정보 수집 및 분석
어플리케이션 플랫폼 관리 도구	어플리케이션 플랫폼에서 기본적으로 제공하는 모니터링 도구로 기본적인 성능 정보를 제공		실시간 기반의 시간당 처리량, 응답시간 등 기본적 성능 정보 수집 및 분석
Profiler	어플리케이션의 워크로드모델링, 런타임 CPU/메모리 사용 패턴, 쓰레드 실행 정보 및 병목지점 분석을 제공	Optimizeit, JProfiler, HP jmeter, .NET CLR profiler 등	JVM 및 CLR을 기반으로 하는 어플리케이션의 실행 정보 수집, 분석 및 튜닝
APM 솔루션	어플리케이션의 운영 상태 모니터링 및 상세한 Profile 정보 제공	Topaz, Wily, i3, APM Suite, Strobe 등	어플리케이션의 전반적 성능 정보 수집, 분석 및 튜닝
End User 성능 관리 도구	분산된 Agent를 통해 주기적으로 최종 사용자 측면의 응답 시간 측정 및 분석 정보 제공	ApplicationVantage 등	최종 사용자 측면의 응답 시간 수집 및 분석
성능테스트 도구	워크로드모델에 따른 테스트 스크립트 작성 및 가상 사용자 기반의 부하 테스트 기능 제공	Loadrunner, SilkPerformer, e Load, QALoad 등	시간당 처리량, 응답 시간 등 수집, 병목 지점 분석 및 용량 관리 지원
로그분석기	워크로드, 어플리케이션 사용량 및 오류/예외 정보 제공	WebTrends 등	워크로드모델, 시간당 처리량, 응답 시간 및 오류 및 예외 분석
코드 Instrumentation	응용 어플리케이션에 관련 코드를 추가하여 특정 실행 정보를 제공		응용 어플리케이션 프로파일링 및 커스텀 성능 지표(예:주문처리 응답 시간 등) 수집

이 중 성능 테스트 도구, 로그 분석기를 제외한 나머지 도구 또는 기법들은 측정 항목, 주기 및 수집 메커니즘에 따라 대상 시스템의 성능에 상당한 악영향을 미칠 수도 있다. 특히, 운영 시스템에 대한 운영중의 성능을 측정하고자 할 경우에는 이들이 자원 사용량 및 정의된 성능 목표 측면에서 어느 정도의 영향을 주는지 사전에 충분히 검토한 후 적용되어야 한다. 성능 측정을 위해 적용되는 각종 도구 및 기법들은 기본적으로 시스템의 자원 사용을 최소한으로 사용하도록 구성하거나 최소한으로 사용하는 제품을 선택하여야 한다.

일반적으로 운영 환경의 경우, 정상 운영 시에는 health check 측면의 데이터 수집 및 모니터링을 수행하고 이상 징후가 감지된 경우(임계치 초과 등)에만 제한된 범위에서 자동으로 상세 데이터 수집을 수행하는 것이 바람직하다. 또는, 정기 유지보수 등을 위한 계획된 가동 중지 시간에 문제가 발생된 영역에 대한 성능 테스트를 수행하거나 별도의 테스트 환경을 마련하여 문제 발생 시점과 유사한(가급적 동일한) 워크로드 모델을 기준으로 독립된 환경에서 성능 테스트를 수행하도록 한다.

만일, 감지된 이상 징후가 비정상적인 급격한 자원 사용(특히, 커널모드(privileged mode)의 CPU 과다 사용)일 경우에는 상세한 성능 측정으로도 문제 발생의 원인을 분석하기 어렵기 때문에 해당 시스템의 운영 체제에서 지원하는 메모리 덤프(crash dump)를 생성하도록 하는 방안도 고려되어야 한다.

(나) 성능 테스트

성능 테스트는 예상되는 평상 시 및 최대(peak)의 부하량 조건에서 해당 어플리케이션이 적절한 수행이 가능하고 용량 증가에 대비하여 충분한 확장이 가능한지를 확인하기 위하여 수행된다. 일반적으로 성능 테스트는 다음과 같이 분류될 수 있다〈표 4-39〉.

〈표 4-39〉 성능 테스트 유형 및 수행내용

성능테스트 유형	목적	수행 내용
부하(load) 테스트	어플리케이션이 정의된 성능 목표를 만족하는 지 확인	평상 시 및 최대(peak)의 부하량에서 어플리케이션의 성능 지표(응답 시간, 시간당 처리량, 자원 사용량 등)를 측정함
과부하(stress) 테스트	어플리케이션이 과부하 상황에서 정상적인 기능이 수행되는 지 확인	최대(peak)치를 초과하는 부하량에서 동기화, 자원 경합 및 메모리 누수 등과 관련된 문제가 발생되는 지 평가함
용량(capacity) 테스트	어플리케이션이 예상되는 사용자 또는 데이터 증가 시에 정의된 성능 목표를 만족하기 위한 필요 시스템 용량을 확인	과거의 성능 측정치를 기준으로 증가된 부하량에서 정의된 성능 목표를 만족하기 위한 필요 시스템 자원 산정 및 확장 전략을 수립함

효과적인 성능 테스트를 위해서는 실 운영 시스템 또는 동일 혹는 축소 규모의 전용 테스트 시스템에 어플리케이션을 구성하고 실 사용자의 작업 유형, 조건 등이 최대한 정확히 반영된 워크로드모델 혹은 테스트 시나리오(테스트 케이스, 데이터, 예상 결과)를 작성하는 것이 필요하다. 이를 바탕으로 성능 테스트를 통하여 다음과 같은 목표를 달성할 수 있다.

- 다양한 수행 조건 하에서 정의된 성능 목표와 관련된 시스템 구성요소(서버, 네트워크, 데이터베이스, 응용 소프트웨어)의 수행 특성을 확인
- 시스템 구성요소의 병목, 예외 및 오류의 발생 여부 확인과 개선 결과 검증
- 시스템 구성요소에 대한 구성 튜닝 및 최적화

성능 테스트를 통해 확인할 수 있는 시스템 구성요소의 수행 특성은 다음과 같다.

- 성능 지표 값 : 응답 시간, 시간당 처리량, 자원 사용량, 지원 가능한 최대 동시 사용자 수 확인
- 성능 지표 추이(변화) 유형 : 다양한 워크로드모델 및 부하량에서의 성능 지표의 추이 또는 변화 유형(예 : 특정 부하량에서 CPU를 일정하게 사용하는지 또는 일정치 못하게 사용하는지 확인)
- Breaking Point : 어플리케이션이 사용자의 요청을 정상적으로 처리하지 못하는 부하량을 확인

- 오류의 증상 및 원인 : 어플리케이션이 사용자의 요청을 정상적으로 처리하지 못하는 경우의 증상 및 원인을 확인
- 용량 확장 방안 : 예상되는 부하량 증가에 대비하여 정의된 성능 목표를 만족시키기 위해 튜닝 또는 용량 확장이 필요한 구성요소 및 그 규모를 확인

세 가지의 성능 테스트 유형 중 부하(load) 테스트와 과부하(stress) 테스트에 대하여 좀 더 살펴보기로 한다.

〈표 4-40〉 부하 테스트 및 과부하 테스트

테스트 유형	준비물	산출물	수행방법
부하(load) 테스트	- 성능 목표 - 워크 로드 모델 - 테스트 시나리오 - 테스트 계획	- 수정된 테스트 계획 - 테스트 결과(최대 부하에서의 자원 사용량 및 최대 지원 가능 동시 사용자 수 포함) - 발견된 병목, 예외 또는 오류 사항 및 개선 권고 내역	평상시의 부하량에서부터 최대(peak) 부하량까지 조금씩 부하량을 늘려가면서 정의된 테스트 시나리오에 대하여 반복적으로 수행
과부하(stress) 테스트	- 부하(load) 테스트 결과물 - 테스트 시나리오 - 예상되는 문제점 - 테스트 계획	- 수정된 테스트 계획 - 테스트 결과 - 발견된 문제점(동기화, 경합 조건, 메모리 누수, 데이터 유실 등) 및 개선 권고 내역	최대(peak) 부하량보다 큰 부하량 조건에서 부하(load) 테스트에서 도출된 시나리오 또는 조합된 시나리오에 대하여 반복적으로 수행

부하(load) 테스트와 과부하(stress) 테스트의 수행 절차는 다음과 같다.

① 주요 업무 시나리오 식별

주요 업무 시나리오는 빈번하게 수행되거나 많은 시스템 자원을 요구하는 업무 기능 중 핵심 업무 기능 즉, 해당 시스템의 성능을 대변할 수 있는 업무 기능을 선정한다. 개발 방법론을 적용하는 경우, 일반적으로 요구사항분석 단계에서 식별될 수 있다. 예를 들어, 온라인 주문의 경우 다음과 같이 선정될 수 있다.

로그인 → 제품 목록 조회 → 특정 제품 검색 → 제품 정보 조회 → 주문 바구니 추가 → 결재 정보 확인 및 주문 처리 → 주문 결과 확인 → 로그아웃

과부하(stress) 시험의 경우에는 부하(load) 테스트를 통해 병목 또는 오류가 발생되었던 시나리오를 중심으로 선정한다.

② 워크로드 특성 식별

각각의 주요 업무 시나리오에 대하여 몇 명의 사용자(동시 사용자 또는 동시 요청자)가 어떤 비율로 단위 업무 기능을 수행할 것인지 정의한다. 예를 들어, 위에서 정의한 온라인 주문 시나리오에 대하여 다음과 같이 정의할 수 있다.

〈표 4-41〉 워크로드 특성 식별(예시)

단위업무	비율(%)	동시사용자	단위작업	수행횟수
제품 정보 조회	60	600	메인페이지 접속	1
			로그인	1
			제품 목록 조회	8
			제품 정보 조회	16
			로그아웃	1
제품 검색 조회	25	250	메인페이지 접속	1
			로그인	1
			제품 목록 조회	4
			제품 검색	6
			제품 정보 조회	12
			로그아웃	1
주문처리	15	150	메인페이지 접속	1
			로그인	1
			제품 목록 조회	4
			제품 검색	2
			제품 정보 조회	8
			주문 바구니 추가	2
			결재 정보 확인	1
			주문 처리	1
			주문 결과 확인	1
			로그아웃	1
계	100	1,000		

과부하(stress) 테스트의 경우에는 부하량을 최대(peak)치 이상으로 늘리는 것 이외에 평상시로 정의된 단위 업무의 비율을 변경하여 수행할 수도 있다. 즉, 위의 예에서 부하(load) 테스트의 경우 제품 정보 조회, 제품 검색 조회, 주문 처리가 각각 60%, 25%, 15% 였으나 과부하(stress) 테스트의 경우에는 각각 10%, 5%, 85%로 변경하여 수행할 수 있다.

③ 성능 지표 식별

정의된 성능 목표 및 예상되는 병목 가능 영역을 고려하여 테스트 수행 중에 측정해야할 성능 지표 및 항목을 선정한다. 테스트를 반복하여 수행하는 과정에서 성능상의 병목으로 판단되는 징후가 발견되었을 경우에는, 관련된 측정 항목을 추가하여 다시 테스트를 수행하도록 한다. 예를 들어, 일반적인 비즈니스 어플리케이션의 경우 정상적인 CPU 사용량은 사용자(user) 모드가 시스템(kernel 또는 privileged) 모드보다 높게 나타나는데 테스트 결과 반대로 나타날 경우에는 시스템 관련 이벤트(예 : 페이징, 컨텍스트 스위치, 인터럽트, I/O, lock 등)를 상세하게 모니터링할 수 있는 측정 항목들을 추가한다.

성능 지표를 식별한 후에는 평상시의 부하량을 기준으로 각 성능 지표에 대한 기준치(baseline metric)를 설정하도록 한다. 설정된 기준치는 다양한 부하량에서의 테스트 결과를 분석하는데 활용될 수 있다.

과부하(stress) 테스트의 경우에는 예상되는 문제점(동기화, 경합 조건, 메모리 누수, 데이터 유실 등)을 고려하여 관련 측정 항목을 부하(load) 테스트에서 사용된 측정 항목에 추가하여 사용한다.

④ 테스트 시나리오 정의

이전 단계에서 정의된 워크로드 특성 및 측정 지표, 항목을 바탕으로 테스트 시나리오를 정의한다. 예를 들어, 위에서 정의한 온라인 주문 시나리오에 대하여 다음과 같이 정의할 수 있다.

〈표 4-42〉 테스트 시나리오 정의 – 부하 테스트(예시)

시나리오 명	온라인 주문(복합)
부하량	동시 사용자 : 500 명 Think Time : 15초 ~ 30초(랜덤) 데 이 터 : 제품 정보 – 10만건
수행시간	30분(또는 1000건 주문 처리)
워크 로드 분포	제품 정보 조회 : 300 명(60%) 제품 검색 조회 : 125 명(25%) 주 문 처 리 : 75 명(15%)
예상 결과	시간당 처리량 : 100 TPS 동시 처리건수 : 50건 평균 응답 시간 : 3 초 자원 사용량 : CPU – 75%, 메모리 – 30% free

과부하(stress) 테스트의 경우에는 다음과 같이 정의할 수 있다.

〈표 4-43〉 테스트 시나리오 정의 – 과부하 테스트(예시)

시나리오 명	온라인 주문(복합) 과부하
부하량	온라인 주문(복합) 과부하 동시 사용자 : 1,00 명 Think Time : 15초 ~ 30초(랜덤) 데 이 터 : 제품 정보 – 20만건
수행시간	2일
워크 로드 분포	제품 정보 조회 : 100 명(10%) 제품 검색 조회 : 50 명 (5%) 주 문 처 리 : 850 명(85%)
예상 결과	시간당 처리량 : 75 TPS 이상 평균 응답 시간 : 10초 이내 기 타 : 데드락이 발생하지 않아야 함 플랫폼 오류율이 10%를 초과하지 않아야 함 데이터 유실이 1%를 초과하지 않아야 함

예상 결과에 기술된 성능 지표 값들은 단지 예제일 뿐이므로 이를 참고로 각 기관의 시스템 특성, 성능 목표 및 테스트 환경에 맞게 변경하여 적용하도록 한다.

⑤ 테스트 수행

정의된 시나리오를 바탕으로 실제 테스트를 수행할 수 있는 환경을 구축한다. 성능 테스트는 RoadRunner, QALoad 등과 같은 전용 테스트 도구를 활용하는 것이 일반적이지만 시스템의 특성이나 테스트 환경에 따라 테스트 도구를 개발하여 사용할 수도 있다. 테스트 도구(또는 개발 프로그램), 테스트 데이터 및 테스트 시스템이 준비되면 정의된 시나리오에 대한 테스트 스크립트를 작성하고 최소 부하량(동시 사용자 1명)에서 작성된 스크립트 및 테스트 환경을 검증한다.

검증이 완료되면 테스트 시나리오에 따라 부하(load) 테스트 및 과부하(stress) 테스트를 수행하는데, 주의해야 할 점은 가상 사용자(Virtual User)를 생성하는 클라이언트 컴퓨터 및 네트워크에서 병목이 발생되지 않도록 충분한 사양 또는 대역폭을 갖추거나 여러 대의 컴퓨터를 준비하도록 해야 한다는 것이다.

J2EE 및 .NET 어플리케이션의 경우 플랫폼의 재기동 후에는 처음 몇 번의 요청에 대해서 JIT 컴파일 및 캐싱 등을 위한 준비 과정을 거치게 된다. 이 기간 동안의 데이터는 수집 또는 분석 대상에서 제외하거나, 1 ~ 2분 정도의 준비(warm up) 테스트를 사전에 수행한 후 원래의 테스트를 수행하도록 한다.

⑥ 결과 분석

각 테스트 시나리오에 대한 테스트 결과 데이터를 예상 결과와 비교하여 위반 사항이 있는지 확인한다. 병목 혹은 오류가 발견된 경우에는, 해당 원인을 분석하고 결과에 따라 개선 작업을 수행한 뒤 동일 테스트를 다시 수행한다.

부하(load) 테스트 단계에서는 일반적으로 정의된 성능 목표를 위반할 때까지 부하량을 증가시켜 테스트하고 과부하(stress) 테스트의 경우에는 시스템의 자원(CPU 또는 메모리)을 최대한으로 사용하는 수준보다 일정 부분 더 초과하는 수준까지 수행한다.

부하량을 조금씩 증가시켜 테스트를 수행한 뒤에는 결과 데이터를 종합하여 부하량에 따른 시간당 처리량, 부하량에 따른 응답 시간 또는 부하량에 따른 자원 사용량 그래프 혹은 표를 작성하도록 하고, 일정 수준 이상의 사용자 요청에 대해서는 거절하도록 구성된 시스템의 경우에는 부하량에 따른 요청 건수 대비 거절 비율도 분석 결과에 포함시키도록 한다.

(3) 분석 방법

성능 병목은 정보시스템의 처리 능력을 감소시키는 자원 혹은 장치를 의미하는 것으로 비즈니스 어플리케이션 시스템의 경우, 대부분 CPU, 메모리, 디스크 I/O, 네트워크 I/O 등과 같은 내부 시스템 자원과 데이터베이스 연결, 네트워크 대역폭 등과 같은 외부 시스템 자원이 해당된다.

모든 시스템의 자원은 제한되어 있기 때문에 성능 병목을 완전히 제거하는 것은 불가능하다. 다만, 대상 시스템에 대한 성능 분석 결과가 정의된 성능 목표를 만족시키기 못할 경우에는, 성능 측면에서 영향이 큰 병목들을 영향도가 작은 다른 자원(병목)으로 옮기거나 약화시켜 전반적으로 정의된 성능 목표를 만족시킬 수 있도록 하는 것이다.

어떤 자원에 의해 성능 병목이 발생하는지를 판단하기 위해서는 성능 측정을 통해 수집된 구간별 응답 시간, 자원 사용량, 시간당 트랜잭션 처리량 및 대기 큐와 관련된 측정 데이터를 중심으로 종합적인 분석이 필요하다. 예를 들어, 일정 시간 동안 CPU의 런 큐(run queue) 사이즈가 CPU 개수 × 2 또는 3을 초과하면 CPU의 병목을 의심할 수 있지만 데이터베이스 커넥션 등과 같은 외부 자원에 대한 대기가 많이 발생하였다면 사용자의 요청이 처리되지 못하고 계속 누적되었기 때문에 발생된 현상으로 판단할 수 있다. 이런 경우에 CPU 점유율은 70 ~ 80% 보다 낮게 되고 결국 CPU의 병목이라고 할 수 없는 것이다.

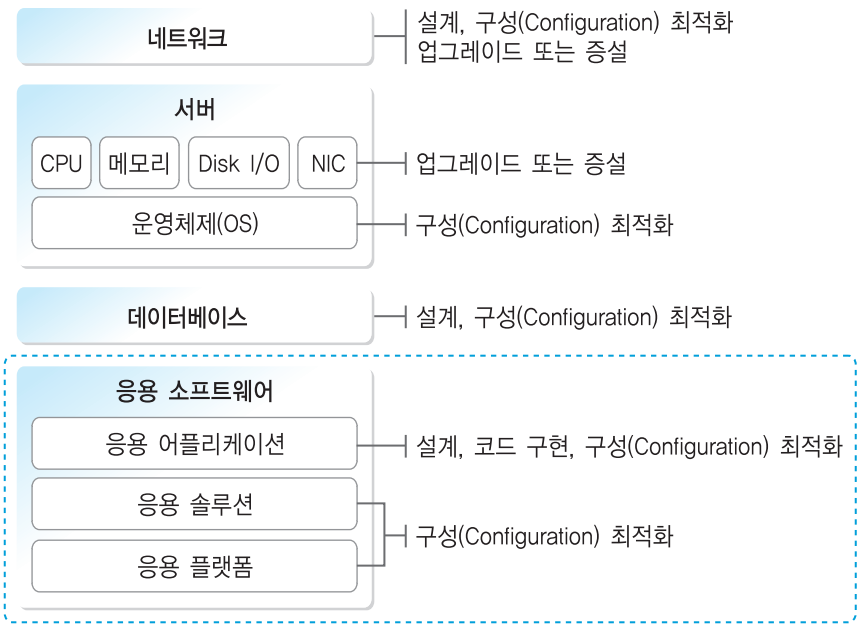
이러한 성능 병목의 근본 원인을 찾아내기 위해서는, 해당 현상에 대한 보다 정밀한 데이터를 수집하여야 한다. 예를 들어, 데이터 분석 결과 GC(Garbage Collection)가 빈번하게 수행(J2EE의 경우 Old generation, .NET의 경우 generation 2)되었을 경우에는 적당한 프로파일러를 사용하여 어플리케이션의 전반적인 메모리 사용 유형을 분석하는 것이 필요하다. 일반적인 웹(Web) 정보시스템의 경우, 가장 흔한 성능 병목의 원인은 다음과 같다.

- 웹/어플리케이션 서버 티어(tier) – 비효율적인 세션 관리, 쓰레드간의 경합, 장시간 실행되는 트랜잭션, 과다한 로깅(logging) 등

- 데이터베이스 서버 티어(tier) – 잘못된 데이터베이스 설계 및 구성, 비효율적인 SQL 및 Stored Procedure 등

나. 성능 개선 방안 수립

성능 측정을 통해 수집된 데이터를 분석하여 성능 병목 또는 문제의 원인이 파악되었으면 이에 대한 개선 방안을 수립하게 되는데, 일반적으로 적용되는 성능 개선의 대상 및 범위는 (그림 4-13)과 같다.



(그림 4-13) 성능 개선의 대상 및 범위

일반적인 웹 비즈니스 어플리케이션의 경우, 응용 소프트웨어와 관련하여 또 다른 중요한 개선 범위 중의 하나는 네트워크, 서버 등과 함께 고려하여야 하는 사항으로 웹서버, 어플리케이션 서버, 데이터베이스 서버를 어떻게 분리 또는 통합할 것인지에 대한 것이다. 이는 전체 아키텍처에 직접적인 영향을 미치는 사항이므로 변경 시에는 상당한 시간과 비용이 필요하다. 따라서, 개발 단계에서 성능 시험 결과 기존 설계 사항을 변경해야 할 경우에만 수행되는 것이 일반적이다.

네트워크, 서버 및 데이터베이스는 본 지침서의 4.2절 분야별 성능관리 프로세스의 해당 분야를 참고하도록 한다.

(1) 대상 선정

일반적으로 성능 개선의 효과가 가장 큰 영역은 응용 어플리케이션이지만 모든 시스템에 동일하게 적용되는 것은 아니다. 따라서, 성능 측정 데이터 분석 결과 도출된 성능 병목 및 문제에 대한 근본 원인을 기준으로 성능에 미치는 영향도를 고려하여 우선 순위를 결정하는 것이 가장 바람직하다.

(2) 방안 수립

성능 개선 방안은 분석된 성능 병목 또는 문제의 근본 원인들을 종합적으로 판단하여 수립한다. 예를 들어, 대기 큐에 쌓이는 클라이언트 요청들을 처리할 쓰레드들의 개수가 부족해서 응답 시간이 길어지는 상황에서 실행되는 전체 쓰레드의 갯수가 많아 컨텍스트 스위치에 따르는 오버헤드로 CPU 사용량이 높다고 하면, 전체 쓰레드 개수는 줄이되 요청을 처리할 쓰레드는 늘려야 하므로 Trade · off를 고려하여 줄일 수 있는 쓰레드들을 선정하여 이를 개선 방안에 포함시켜야 한다.

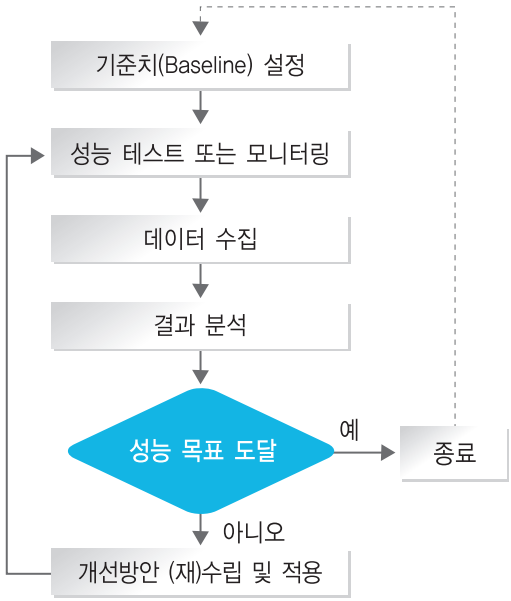
각 측정 항목과 관련된 개별적인 개선 방안에 대하여는 '가. 데이터 수집 및 분석' 의 '(1) 측정 항목' 을 참고할 수 있다.

성능 개선 방안에는 구체적인 개선 항목과 Trade Off, 예상되는 개선 효과, 우선순위, 개선 항목 수행에 소요되는 시간 및 인원, 기타 고려사항들을 기술하도록 한다.

다. 성능 개선 실행

(1) 성능 조정(튜닝)

성능 조정(튜닝)은 개발 및 운영중인 시스템에 대한 성능 테스트 또는 운영 상태 모니터링의 결과가 정의된 성능 목표를 만족시키지 못하는 것으로 나타나는 경우에 성능 병목들을 식별하여 우선순위에 따라 원인을 분석하고 개선 방안을 마련하여 체계적으로 적용함으로써 궁극적으로 정의된 성능 목표를 만족시킬 수 있도록 하는 활동이다. 성능 조정(튜닝)의 절차는 다음과 같다(그림 4-14).



(그림 4-14) 성능 조정(튜닝) 절차

기준치(baseline) 설정은 성능 목표의 정의 또는 정의된 성능 목표의 확인과 운영 상태 모니터링 또는 최초의 성능 테스트를 통해 수집된 성능 지표들의 측정값, 그리고 기준 시나리오(워크로드 모델)를 정의 또는 확인하는 작업이다.

성능 테스트 또는 모니터링 및 데이터 수집단계의 최초 수행 시점이 기준치(baseline) 설정 시점과 다를 경우(차이가 많을 경우)에는 반드시 기준치 설정 이후에 성능 조정(튜닝) 대상 시스템의 구성 요소(특히 어플리케이션, 플랫폼 소프트웨어 버전 및 파라미터)에 대한 변경이 없었는지를 확인하여야 한다. 이 단계에서부터 결과 분석까지는 4.2.4절, 응용 소프트웨어의 '가. 데이터 수집 및 분석'을 참고하도록 한다.

결과 분석 후 개선방안 (재)수립 및 적용에서는 4.2.4절 응용 소프트웨어의 '나. 성능 개선 방안 수립'을 참고로 수립된 개선방안들 중 우선순위가 높은 것부터 적용하되 하나의 개선 방안 내에 여러 개의 독립적인 개선 항목이 있는 경우에는 우선순위에 따라 반드시 한번에 하나의 개선 항목만을 적용하도록 한다. 예를 들어, 개선 항목으로 응용 어플리케이션 코드 변경과 메모리 관련 운영체제 파라미터 변경이 선정되었을 경우, 응용 어플리케이션 코드 변경의 우선순위가 높다면 첫번째 단계로 응용 어플리케이션의 코드를 변경하여 그 결과를 확인하고 분석한 뒤 필요 시 메모리 관련 운영체제 파라미터 변경을 수행하도록 한다.

개선 항목을 적용한 뒤에는 다시 성능 테스트 또는 모니터링을 통해 성능 데이터를 수집하고 이후 분석하는 과정을 반복하게 된다. 경우에 따라서는, 새로운 개선 방안 또는 항목이 추가될 수도 있고, 개선 항목 적용 이전에 발견되지 않았던 새로운 성능 병목 또는 문제가 발견될 수도 있는데, 개선 결과를 평가할 때에는 정의된 성능 목표를 기준으로 개선되었는지 아닌지가 보다 중요하다고 할 수 있다.

분석 결과, 정의된 성능 목표가 만족되었을 경우에는 성능 조정(튜닝)을 종료하게 되고 이 때의 측정값 및 환경 구성 정보(파라미터 등)를 새로운 기준치(baseline)로 관리한다. 성능 조정(튜닝)은 제한된 자원(비용, 시간, 환경 등) 내에서 수행되는 작업이므로 응답 시간 3초를 목표로 하는 시스템에 대하여 1초의 응답 시간을 달성하려고 하는 것은 바람직하지 않다. 또한, 성능 조정(튜닝) 작업으로 모든 응용 어플리케이션에 대하여 정의된 성능 목표를 만족시키기 어려울 수도 있다. 예를 들어, 응용 어플리케이션 3개의 성능 목표가 모두 응답 시간 3초인 시스템에 대하여 다음과 같이 측정 되었다고 하자.

〈표 4-44〉 어플리케이션 튜닝 전 · 후의 응답시간 비교(예시)

어플리케이션	사용률	응답시간(튜닝 전)	응답시간(튜닝 후)
A	35%	3.5초	2.8초
B	60%	5초	2.5초
C	5%	2초	3.5초

위의 경우, 성능 조정(튜닝) 결과 모든 응용 어플리케이션이 정의된 성능 목표를 만족시키지는 않지만 가장 많이 사용되는 A, B 어플리케이션에 대하여는 성능 목표를 만족하고 있다. 따라서, 성능 개선의 효과측면에서 위와 같은 결과에 대해서도 성능 조정(튜닝)을 종료할 수 있는 조건이라고 할 수 있다.

(2) 용량 증설

응용 소프트웨어의 용량 증설은 서버 및 네트워크의 용량 증설을 필수적으로 요구하게 된다. 웹 정보시스템의 경우, 가용성, 관리성 및 기술적 제약사항 등을 고려하여 웹서버는 수평 확장을, 백엔드(back end) 서버는 수직 확장을 하는 것이 일반적이라고 할 수 있다.

서버 및 네트워크의 용량 증설에 대하여는 본 지침서의 4.2절, 분야별 성능관리 프로세스의 해당 분야를 참고하도록 한다.

4 3 검증 및 결과 관리

4 3.1 개선효과 검증

개선 효과를 검증하기 위해서는 전체 개선 작업량에 대한 분석 효용을 검증하는 것이 일반적이다. 이는, 성능 개선 적용 대비 성능 개선 비율을 검증하여, 성능 조정 목표 대비 성능 개선 상황의 효율성(Efficiency)을 판단하는 것을 말한다.

〈표 4-45〉 성능관리 개선효과 검증(예시)

구 분	내 용
측정지표	성능개선 대상 선정 건수
정의	성능개선 대상 (튜닝 적용) 건수
관련절차서	성능 관리 절차서에 사례 반영
계산 방법	(Σ성능개선 대상 적용 건수 / Σ성능개선 목표 건수)×100
데이터 수집자	운영파트
데이터 분석자	운영파트
측정주기	반기 1회
보고주기	반기 1회
특기사항	성능개선 항목 수는 월 성능 분석 결과 도출된 성능개선 항목으로 계산

성능 개선 효과를 검증하는 방법은 양적인 검증보다는 실제 서비스를 이용하고 있는 사용자로부터의 피드백을 통하여 질적인 개선 효과를 검증하는 것이 바람직하다.

가. 사용자로부터의 검증

IT 인프라의 업무 응용 어플리케이션을 사용하는 사용자의 반응을 통하여 업무 목표의 달성 여부를 검증하는 방법이야말로, 실제 인프라를 구축한 목표와 일치하게 되므로 가장 명확한 개선 효과 검증의 척도가 될 수 있다. 따라서, 각 사용자가 대 고객 서비스를 처리하면서 느끼는 만족도나 사용자 만족도를 계수화 한 SLA에 비교하여 개선 효과를 검증하는 방안을 마련하여야 한다.

세부적으로 작성된 설문 항목을 개별 사용자에게 질의하는 시스템을 자동화된 모니터링 환경에 추가하여 작성한다면 사용자의 성능 개선 후의 만족도를 즉각적으로 반영할 수 있을 것이다.

업무 목표를 세부적인 수치로 환산하여 마련한 임계치에 비례하여, 임계치를 위반한 값이 정상화 되었는지 여부를 판단하는 것과 정상적인 서비스 상황하에서의 기준치(Baseline)를 비교 판단하여 현재의 성능 개선 후의 상황을 검증하는 방법은 명확한 수치화가 동반되어야 하므로 성능 측정 환경 구현 시 철저한 분석과 설계 및 구현이 수반되어야 한다.

나. 연계분야로부터의 검증

성능 개선 효과는 운영상태관리와 서비스데스크관리, 장애관리, 용량관리 연계 분야로부터 직/간접적인 피드백을 받을 수 있다. 운영상태관리의 성능 저하 문제는 임계치의 Critical이나, Warning 알람이 사라지게 됨으로써 직접적인 피드백을 받을 수 있고, 서비스데스크관리 및 장애관리 분야를 통하여 해소된 사용자들의 성능 문제에 대한 보고를 받을 수 있도록 연계성을 유지하여야 한다. 또한 성능관리 분야에서 이관되어 용량증설을 통한 개선 효과의 결과 검증은 용량관리 분야로도 보고 되어져야 하며, 만일 성능 개선 효과에 대한 이면적인 여러 개선 기준이 변경된 경우, SLA의 판단 기준이 변화될 수도 있을 것이다. 각 운영분과와의 연계를 통한 검증은 연계분야와의 관계는 본 지침의 3.4절, 연계 분야를 참조한다.

4 3.2 결과 관리

성능관리 상의 모든 절차는 문서로 기록되고 관리되어야 하며, 모니터링되고 분석된 데이터는 구성관리 데이터베이스에 저장되어 모든 연계 분야에서 효율적으로 데이터들이 활용될 수 있도록 관리되어야 한다. 성능관리의 각 단계(계획, 데이터 수집, 분석, 조정)에서 기록 및 관리되어야 할 내용들은 다음과 같다.

가. 성능관리 계획 단계 : 성능관리 대상 및 방법이 명시

- 성능분석 요구(의뢰)서 및 성능조정 요구(의뢰)서
- 서비스 현황 분석 문서
- SLA 관련(임계치 설정) 데이터
- 사용자 만족도의 수치화(계측화)
- 성능관리 대상 및 측정 항목
- 측정 항목별 임계 값 및 임계 값 초과 항목에 대한 운영상태관리 여부

나. 성능 데이터 수집 및 분석

: 성능 측정 대상별, 항목별 측정 방법 및 주기, 분석 방법에 대한 기록관리

- 성능 측정 항목에 대한 측정 방법 및 측정 주기
- 성능 측정 결과를 이용한 성능 분석 주기
- 성능 분석 대상의 구성 정보(대상명, 설치 위치, 구성 항목 등)
- 성능 측정 도구 및 측정 방법 정보(측정 주기/기간, 측정 항목 리스트)
- 성능측정 결과에 대한 성능 및 추이 분석
- 통합된 관점의 성능 분석 결과(보고서 형태로 관리)

다. 성능 조정 및 연계분야 관리

: 성능 분석 결과를 근거로 한 성능 조정(튜닝) 대상의 선정 및 조정(튜닝) 실시

- 성능 조정 대상 선정 근거
- 성능 조정 방법에 대한 기록(방법론 등을 세부 절차 기록)
- 성능 조정 적용 방안
- 성능 조정 전 개선 활동의 영향도 및 효과 예측
- 성능 조정 결과 검증 방법
- 연계분야로의 영향도 분석

라. 성능관리 결과 처리 시 고려사항

▶ 데이터는 정확하게 분석되었는가?

: 가능한 한 데이터를 정확하고 분명하게 분석하고 조사하는 작업을 그 무엇보다 중요하게 다루어야 한다. 데이터에 근거하지 않고 추측에 근거한다면 올바른 결론을 낼 수 없다.

▶ 간결한 리포트 작성

: 보통 성능 분석 리포트를 작성할 때 분석에 대해 장황하고 세밀하게 정리하는 것을 지양하고 간결하고 명확하게 작성해야 한다.

▶ 성능조정에 대한 최종결정권자

: 성능 분석 리포트 및 조정(튜닝)계획서를 작성하여 최종결정권자에게 프레젠테이션을 하거나 리포트를 제출할 때, 그 결과에 대한 수많은 질문을 받게 될 것이다. 사전에 제기될 만한 질의에 대한 답변을 정리해 두고, 보다 빠른 결정을 할 수 있도록 준비해야 한다.

▶ 다양한 시각에서의 성능분석/관리

: 현재 분석/조정, 관리된 성능보고서와는 다른 방향에서 성능상태를 바라보는 시각이 있을 수 있다. 만약 성능 분석 결과에 대하여 관련된 연계분야나, 최종결정권자가 다른 의견을 제시했을 경우에는 이에 대해서 반대 의견만을 제시해서는 안 된다. 정보시스템이 매우 복잡하게 구성되어 있으며 여러 다른 상황과 밀접하게 맞물려 있기 때문에 다양한 시각에서 정보시스템을 분석하고, 다양한 사용자의 요구사항에 대하여 분석/조정, 관리할 필요가 있다.

▶ 비용 효율적인 성능 분석 실시

: 오랜 시간에 걸쳐 복잡한 분석 기법으로 성능을 분석하려고 많은 시간과 비용을 소비하기도 한다. 그러나 복잡한 기법을 분석하는 것도 중요하지만 시스템 관리에 있어서 오랜 경험을 바탕으로 기본에 충실한 성능 분석 기법을 활용한 이용하는 것으로도 훌륭한 성능 분석 결과를 도출해 낼 수 있다

4 3.3 KPI(Key Performance Indicator)의 적용

성능관리 프로세스가 성공적으로 수행되기 위한 핵심 성공요소를 확인하고 그 성공요소를 지원하는데 있어 프로세스 이행 성과를 평가하기 위한 주요성과지표를 도출하여 평가하도록 한다. 이러한 프로세스 성능 평가를 통해서 프로세스 이행의 미비한 사항을 개선하고 적용함으로써 프로세스가 지속적인 발전을 해 나갈 수 있다.

주요성과지표는 측정 가능하고 현실적인 지표, 사용자에 대한 고품질의 서비스를 지원할 수 있는 지표로 선정하도록 한다. 프로세스의 성숙도가 높아질수록 IT자원 중심의 지표보다는 사용자 중심의 지표가 선정되어야 한다.

또한 선정된 측정지표에 대한 명확한 목적과 정의, 산출방법 등을 반드시 기술하여 평가의 객관성이 확보되도록 한다.

〈표 4-46〉은 일반적인 성능관리의 성공요소 및 주요성과지표들의 예시이며, 각 기관들은 해당기관의 업무 특성에 따라 성능관리 프로세스의 주요성과지표를 도출하여 활용하도록 한다.

〈표 4-46〉 성능관리 성공요소 및 주요 성과지표(예시)

핵심성공요소	성과지표
정확한 성능 요구사항 분석	– 시기 적절한 자원요구 예측 생성 – 자원 사용 추세를 정확히 예측 – 적절한 비즈니스 계획과 용량 계획의 조합 – 비즈니스 계획과 용량 계획의 불일치 건수 감소
현재와 미래의 IT기술에 대한 지식	– 모든 서비스와 컴포넌트들에 대한 성능과 처리량을 모니터하기 위한 능력 증대 – 비즈니스 요구(시간, 비용, 기능)에 맞는 신기술 구현 – 기존 기술 성능과 관련된 문제로 인한 SLA 위반 건수의 감소
비용 효율성	– 성능 유지를 위한 비용 최소화 – 초과 용량산정 억제 – 투자비용의 정확한 예측
성능 요구사항을 충족하기 위한 성능관리 수행 능력	– 성능 저하로 인해 발생된 사고 건수의 감소 – 부족한 용량 또는 성능으로 인해 SLA 위반 건수의 감소 – 성능 분석의 결과로써 만들어진 권고사항(용량증설) 수행의 성공율

5 통합적 성능관리

5 1 구축 절차

통합적 성능관리시스템 구축은 다음과 같이 3단계로 나누어 추진한다.

▶ 구축계획 수립 단계
: 구현 옵션 선정, 적용시스템 선정, 벤더 선정, 상세 구축계획 수립

▶ 구축 1단계
: 통합적 성능관리시스템 기반 구축 및 장애감시/성능관리 역량 구축

▶ 구축 2단계
: 통합백업/스토리지 역량 증대 및 통합 모니터링 역량 확대 구축, 서비스수준 및 운영현황 역량 구축(워크플로우 기능 포함)

통합적 성능관리시스템 추진의 3단계에 대한 상세한 설명은 다음과 같다.

가. 구축계획 수립 단계

구축계획 수립단계에서는 상세 구축계획을 수립하는 단계로서 구현 옵션 선정, 적용시스템 선정, 벤더 선정, 상세 구축계획을 수립한다.

▶ 구현옵션 선정
: 추진 시나리오를 검토하고 추진할 시나리오를 선정한다.

▶ 적용시스템 선정

: 본 지침서의 3장에서 언급한 3.2.5절, '나. 성능관리의 범위' 의 '적용대상 선정기준' 부분을 참조하여 통합적 성능관리 대상 시스템을 선정한다. 적용시스템을 선정하기 위해서 각 데이터센터 또는 공공부문별로 운영되고 있는 시스템 상세 정보를 수집, 분석해야 한다.

▶ 벤더선정

: 유지보수를 제외한 1, 2, 3단계 통합성능관리시스템 구축 프로젝트에서 활용될 솔루션을 선정한다.

▶ 상세 구축계획 수립

: 상세 구축계획 수립에는 다음과 같은 항목들에 대해서 정의한다.

- 범위 : 선정된 구현옵션과 적용시스템을 바탕으로 운영대상, 기능 범위에 대해서 정의한다.
- 구축 일정 : 각 단계별 상세 구축 일정을 수립한다.
- 프로젝트 조직 : 각 단계별 프로젝트를 진행할 프로젝트 조직 구성 및 역할을 수립한다.
- 자원 및 비용 예측 : 각 단계별 투입 자원 및 비용을 수립한다.
- 기타 : 위험관리계획 수립, 아키텍처 정의(개발, 실행, 운영)

나. 구축 1단계

구축 1단계에서는 모든 영역에 대하여 통합적 성능관리시스템을 위한 역량을 구축한다. 시스템 구축을 위한 아키텍처를 확정하고 성능관리도구의 확대구축을 포함한 통합모니터링 역량의 구축, 기능 확장, 서비스수준관리를 포함한 통합운영현황관리 시스템의 구축이 1단계에 포함된다.

▶ 통합운영시스템 설계

: 통합운영시스템의 아키텍처를 설계하고 각 운영도구들이 개발된 통합 아키텍처에 적합하게 구축될 수 있도록 통합구축 가이드를 개발한다.

▶ 통합모니터링 및 성능관리도구 확대구축

: 통합이벤트 대상은 장애감시/성능 이벤트이며, 독립추진도구에 대한 이벤트 통합은 각 도구가 2단계에서 구축·완료되므로 2단계에서 통합한다. 통합대상 도구범위는 기존 사용 중인 운영관리도구와 1단계에서 구축되는 운영관리도구가 대상이다.

▶ 유지보수를 통한 기능확장 대상(장애/변경/자산/구성/사용자 ID관리)

: 해당 운영기능에서 기능개선이 필요한 요구사항에 대하여 유지보수 및 소규모 개발을 통하여 기능을 확장한다.

▶ 통합운영관리 요건정의

: 서비스수준 및 운영현황관리 시스템의 요건 및 의사소통 성격의 워크플로우 요건을 정의한다. 이 요건에 따라 SLM벤더 파일럿(Pilot)을 수행한다.

▶ SLM벤더 파일럿

: 서비스 수준 및 운영현황 솔루션은 비교적 최근에 개발되었으며, 적용사례 또한 많지 않아 솔루션 검증이 요구된다. 솔루션 검증을 위해서 벤더별 파일럿을 수행하며, 솔루션 적용과 custom 개발을 고려하여 벤더를 선정한다. 그리고 선정된 솔루션에 적합한 상세 구축계획을 수립한다.

▶ 서비스수준 및 운영현황관리 시스템 구축(워크플로우 기능포함)

: 기존 운영도구와 1단계에 구축되는 운영도구들 포함한 서비스수준 및 운영현황 역량을 구축한다.

▶ 개별도구 역량구축(독립추진 모듈)

: 통합적 성능관리 프로젝트와 독립적으로 통합백업, 스토리지, Job스케쥴링 도구에 대한 구축을 수행한다.

다. 구축 2단계

구축 2단계에서는 1단계에서 구축된 통합적 성능관리시스템 및 운영현황관리 시스템에 독립 추진된 도구들을 통합하는 작업을 수행한다. 즉, 통합백업, 스토리지, job스케쥴링 도구에 대하여 필요 이벤트 및 정보를 통합할 수 있는 역량을 구축한다.

▶ 독립추진도구를 위한 이벤트 통합
: 구축 2단계 기간중에 구축·완료되는 독립추진모듈(통합백업, 스토리지, job스케쥴링 도구)에 대한 필요이벤트를 통합운영 관리시스템으로 통합할 수 있는 역량을 구축한다.

▶ 독립추진도구를 위한 운영현황관리 정보통합
: 구축 2단계 기간중에 구축·완료되는 독립추진모듈(통합백업, 스토리지, job스케쥴링 도구)에 대한 필요정보를 통합운영 현황관리 시스템으로 통합할 수 있는 역량을 구축한다.

5.2 구축 방안

가. 요구사항

통합적 성능관리 시스템 구축시 요구사항은 정의된 각 운영관리 프로세스가 효과적으로 수행할 수 있도록 도구 측면에서 자동화도구 활용이 필요한 요건들을 정의하였다. 분석은 자동화도구 통합 요구사항 측면과 기능별 자동화도구 요구사항 측면으로 정의하였다.

▶ 자동화도구 통합 요구사항 : 통합운영관리 시스템의 통합 기반이 필요한 정보(이벤트, 데이터, 업무 프로세스) 통합, 기술 통합, 운영환경 관리, 기타로 분류하여 정의하였다.

▶ 기능별 자동화도구 요구사항 : 자동화도구 기능이 필요한 기능별과 인터페이스가 필요한 기능으로 분류하여 정의하였다.

(1) 자동화도구 통합 요구사항

〈표 5-1〉 통합 성능관리 자동화도구 요구사항(정보통합)

통합분류	요구사항
정보통합	▶ 이벤트 통합 기반 효과적인 이벤트 수집 기반이 제공되어야 하며, 여러 성능관리 도구의 모니터링 기능에서 수집된 이벤트를 중앙 콘솔에 제공하여 통합관리 할 수 있어야 함 – 성능관리 도구 – 모니터링 도구 – 백업/복구관리 도구 – Job 스케쥴링 도구 – 스토리지 관리 도구 – 용량관리 도구 – 보안관리 도구 ▶ 이벤트 관리 요건 다음의 문제들에 대한 지원 기능을 제공해야 함 – 실시간 이벤트 처리에서의 시간 지연 방지가 요구됨(기존 관리도구와 신규 관리도구의 연동에 따른 실시간 이벤트 데이터의 시간 지연 문제 발생) – 기존 관리도구와 신규 관리도구에서 각각 수행되는 이벤트 필터링의 단일화가 요구됨
	▶ 데이터 통합 기반 효과적인 데이터 수집 기반이 제공되어야 하며, 여러 운영관리 도구에서 생성/수집되는 데이터 중에서 서비스수준 평가와 운영현황 파악을 위한 데이터를 통합관리 할 수 있어야 함 – 성능분석 도구 – 용량분석 도구 – 스토리지 관리 도구 – 구성관리 도구 – 자산관리 도구 – 장애관리 도구 – 서비스문의/요청관리 도구 – 보안관리 도구

<table>
<tr><td rowspan="2">정보통합
(계속)</td><td>

▶ 데이터 관리 요건

다음의 문제들에 대한 지원 기능을 제공해야 함

– 지원 가능한 기존 관리도구의 데이터 수집 방법 (logfile, API, SNMP, Database Trigger 등)
– 데이터 생성/수집에 대해서 타 운영기능에 필요한 정보를 자동으로 전달할 수 있어야 함
– 기존 관리도구와 신규 관리도구의 Agent가 동일한 성능 데이터를 중복 수집함으로써 불필요한 시스템 부하를 발생할 경우 이에 대한 방안(예: 신규 관리도구의 에이전트의 서버 성능 항목 중 CPU성능 데이터 수집을 inactive할 수 있는지 여부)

</td></tr>
<tr><td>

▶ 업무 프로세스 통합 기반

업무 프로세스 통합이 필요성이 있는 기능들이 다음과 같고, 이를 위해 다양한 도구에서 수행되는 태스크를 연계할 수 있는 기반이 제공되어야 함

– 장애관리 도구
– 자산관리 도구
– 서비스수준관리 도구
– 변경관리 도구
– 구성관리 도구
– 서비스 문의/요청 도구
– 보안관리 도구: ESM

▶ 업무 프로세스 통합 요건

– 통합 모니터링과 보안관리와의 연계
• 모니터링에서 탐지된 보안 이벤트의 조치를 위하여 보안이벤트 발생 및 1차 처리 결과를 보안관리 도구에 해당 이벤트에 대한 사후관리 업무를 생성함
– 장애 관리와 서비스 문의/요청관리와의 연계
• 장애원인과 해결현황, 장애처리시간, 심각도 등의 정보가 서비스 문의/요청관리로 제공되어 응대업무에 활용되어야 함.
– 서비스 문의/요청관리와 장애관리와의 연계
• 서비스 문의/요청에서 장애에 해당하는 문의/요청 접수시 장애관리 도구에 이를 할당하고 자동으로 장애 티켓을 생성할 수 있어야 함
• 장애관리도구에서 장애처리에 따른 장애 처리 상태 정보를 서비스 문의/요청에 연동하여 서비스데스크 요원은 이를 조회 가능 하여야 함

</td></tr>
</table>

정보통합 (계속)	– 통합모니터링과 장애관리와 연계 • 통합 모니터링 콘솔은 장애발생 이벤트에 대하여 장애관리 도구에 장애티켓을 생성하여 후속 장애조치 업무가 신속히 수행될 수 있어야 함 – 변경관리와 구성관리와의 연계 • 작성된 변경계획서에 대하여 설정된 구성관리자에게 전송되어 검토 및 피드백 수렴절차를 자동화하여 변경에 대한 의사소통이 효율화 되어야 함 – 장애관리 프로세스의 효율화 • 장애 조치가 1차 지원선에서 불가능한 경우에 2차 지원그룹으로 장애 내역을 이관하는 프로세스의 자동화 – 성능관리 프로세스의 효율화 • 작성된 성능 개선 보고서가 설정된 어플리케이션 담당자 및 성능책임자에게 전달되어 후속업무가 효율적으로 진행되도록 함 – 용량관리 프로세스의 효율화 • 수립된 용량계획보고서를 설정된 어플리케이션 담당자에게 제공하여 용량증설 또는 계획 검토 등의 후속업무가 효율적으로 진행되도록 함 – Job 스케쥴링 프로세스의 효율화 • 작성된 Job스케쥴 및 보고서가 설정된 관련 담당자들에게 제공되어 후속업무가 효율적으로 진행되도록 함 – 스토리지관리 프로세스의 효율화 • 수집된 스토리지 보고서를 설정된 어플리케이션 담당자 및 스토리지 책임자에게 제공하여 후속업무가 효율적으로 진행되도록 함 – 변경관리 프로세스의 효율화 • 작성된 변경계획서에 대하여 설정된 관리자에게 전송되어 승인/검토 및 피드백 수렴 절차를 자동화하여 변경에 대한 의사소통이 효율화 되어야 함

① 예를 들어 UNIX 성능 분석 도구에서 수집한 이벤트를 모니터링 도구에 이벤트를 전달할 수 있어야 함. 이벤트에 의한 기능간 인터페이스는 각 기능별 요구사항에 정의되어 있음

② 예를 들어 UNIX 성능 분석 도구에서 수집한 성능 정보를 용량관리 도구에 제공하여 용량분석을 수행할 수 있도록 해야 함. 데이터의 기능간 인터페이스는 각 기능별 요구사항에 정의되어 있음.

③ 예를 들어 변경관리에서 서버의 용량 증설 후에 관련 정보를 자산관리와 구성관리에 자동으로 전달할 수 있어야 함.

④ 단순한 정보의 제공, 보고 및 요청 등의 프로세스 통합 요건은 기능별 상세 기능 요건정의에서 기술함

〈표 5-2〉 통합 성능관리 자동화도구 요구사항(기술통합 등)

통합분류	요구사항
기술 통합	▶ 오픈 및 표준 아키텍쳐 지원 - 서로 다른 언어로 개발된 어플리케이션 간의 통합을 제공해야 함 (예를 들면, 자체 개발된 AMS가 C로 개발된 경우 웹으로 연동방안 또는 장애관리 Script와 변경 관리 도구와의 통합 방안 등) - SNMP, CORBA, WBEM(Web-based enterprise management or WMI), JMX 지원 여부 - 타 솔루션의 Manager와 Agent와의 통신을 지원해야 함 - 3계층 관리시스템 지원 시 Manager와 Mid-Level Manager와의 통신을 지원해야 함
	▶ 데이터 보안 및 방어벽 - 중앙 콘솔로 보내지는 데이터에 대한 암호화가 가능해야 함 - 방화벽(firewall)을 통해서 연동이 가능해야 함
	▶ 데이터 압축 - 중앙 콘솔로 보내지는 데이터에 대한 압축이 가능해서 전송 속도의 효율성을 높일 수 있어야 함
	▶ 용이한 커스터마이징 지원 기존 운영환경에 요구되는 기능을 용이하게 커스터마이징하여 추가할 수 있어야 함. 그러기 위해서는 다음과 같은 사항이 지원되어야 함 - 타 시스템과의 인터페이스를 위한 데이터 포맷 변경 및 컨텐츠의 한글화 - 전체적인 GUI(전 UI의 한글화/Look and Feel/그래프 형식 등) - 전산자원 대신 업무시스템으로 표현한 구성 맵 제공 - 이벤트 통보에 다양한 룰 적용(시스템 별/담당자 별/시간대 별) - 사용자 정의 보고서 및 보고서 포맷 (MS Excel, HTML 출력 기능) - In-house 어플리케이션의 성능정보 수집 및 추이분석
운영도구 환경관리	▶ 시스템의 가용성 관리 콘솔 또는 서버 시스템이 다운되는 경우에 복구 또는 재시작이 가능해야 함. 관리서버는 H/A를 지원하여야 함
	▶ Agent상태 모니터링 및 제어 기존관리도구의 Agent 상태 모니터링 및 기동/정지/일시중지 등의 기능을 통합관리도구에서 수행 할 수 있어야 함

운영도구 환경관리 (계속)	▶ 통합 콘솔 모든 플랫폼 관리를 위한 중앙 집중식 논리 콘솔을 제공하고 GUI형태로 C/S 환경, WEB환경, Mobile환경에서도 사용할 수 있어야함
	▶ 운영 환경 뷰 제공 운영 환경을 구성하고 있는 관리 대상, 관리 시스템의 운영 환경 뷰를 제공해야 함. 그리고 상위 수준 뷰와 상세 수준 뷰를 각각 제공해야 함
기타	▶ 지원 플랫폼 DBMS의 종류(Oracle, Sybase, Informix, UDB, MS-SQL), 운영 중인 응용시스템의 종류(Mainframe(OS/390), UNIX(SUN, HP, AIX, OpenVMS, True64 UNIX), Tandem, NT)와 관계없이 관리가 가능해야 하며, 서버에서 메인프레임, 네트워크에서 응용시스템, 분배에서 조작에 이르기까지 전 관리 영역을 지원 해야 함
	▶ GUI 콘솔 사용의 편리성 사용자의 기호나 환경에 맞게 관리환경의 구성 요소인 메뉴, 아이콘, 화면 구성 등의 조정이 가능해야 함 – 직관적인 User Interface와 Navigation Mechanism을 지원해야 함 – 각각의 Administrator의 다른 profiles를 제공해야 함(예, 보안 Admin)
	▶ 3계층 관리시스템 (Manager/Mid-level manager/Agent) 성능정보는 mid-level manager에서 수집하여 일정 주기로 manager로 전송하고, 긴급한 장애 이벤트는 실시간으로 manager로 전송 가능 해야 함

정보 통합 요건에서 파악된 프로세스 통합 요건은 자동화 추진을 위한 고려를 통하여 두 가지 정도의 유형으로 분류될 수 있으며, 각각의 유형에 대한 정의 및 설명은 다음과 같다.

〈표 5-3〉 프로세스 통합 유형

프로세스 통합유형	설명
의사소통의 효율화 (단순 정보보고, 업무 요청, 공지)	– 해당 태스크를 통하여 생성된 정보(현황정보, 분석 보고서 등)를 그 다음 태스크 수행자에게 효율적으로 전달하는 경우 – 업무 요청, 협조 요청 등의 요청 업무 자동화를 요하는 경우 – 생성된 정보를 필요 태스크 수행자에게 공지하는 정도의 프로세스 통합이 요구되는 경우
업무의 흐름 통합 (태스크 생성, 할당, 트래킹)	프로세스 통합을 통하여 수행된 태스크 수행자는 차기 태스크의 생성 및 할당이 가능하며 관련자, 관리자 및 타 태스크 수행자는 각 태스크의 수행상태를 모니터링 함으로써 전체적인 프로세스의 진행을 관리할 수 있도록 함

(2) 기능별 성능관리 자동화 도구 요구사항

〈표 5-4〉 기능별 성능관리 자동화 도구 요구사항

요구사항		
기능	성능 모니터링	▶ 성능 분석 요청 (Workflow 기능 활용) 시스템 성능 분석을 요청할 수 있는 기능이 제공되어야 함
		▶ 어플리케이션, 데이터베이스, 네트워크, 서버의 구성 요소에 대한 성능 모니터링 기능 – 어플리케이션 성능 모니터링 : 성능 분석 대상 측면에서 C/S, Web, WAP 등의 필요한 어플리케이션 성능 측정이 가능해야 함 – DB 성능 모니터링 : 성능 분석 대상 측면에서 Oracle, Sybase, UDB등의 필요한 DB 성능 측정이 가능해야 함 – 네트워크 성능 모니터링 : 성능 분석 대상 측면에서 WAN 간의 네트워크 성능, LAN 사이의 네트워크 성능, Production 시스템 간의 네트워크 성능 등의 필요한 네트워크 성능 측정이 가능해야 함 – 서버 성능 모니터링 : 성능 분석 대상 측면에서 UNIX, NT 서버의 필요한 성능 측정이 가능해야 함(예, UNIX CPU, 메모리, 디스크에 대한 성능 측정)
		▶ 임계값 기능 어플리케이션, 데이터베이스, 네트워크, 서버의 구성요소에 대하여 임계값을 설정하고 이에 해당하는 이벤트 관리가 가능해야 함
		▶ 필터링 기능 심각도 정도에 따라 해당 이벤트 분류가 가능해야 함
		▶ 공지 기능 이벤트와 필터링에 따라 이벤트 정보를 해당 담당자에게 SMS/E-mail 등으로 공지할 수 있는 기능이 있어야 함
	성능 정보 (이벤트/ 데이터) 수집	▶성능 정보 수집 성능 항목을 설정한 주기에 따라 성능 정보를 수집할 수 있어야 함
		▶ 이벤트 관련 성능 정보 제공 성능 이벤트에 대한 관련 이슈 파악을 위해서 발생한 성능 이벤트 수집 시에 관련 성능 정보를 함께 수집/분석할 수 있어야 함
		▶ 성능 모니터링 사용 성능 테스트 및 파악 성능 모니터링에 의해 production에 미치는 성능 정도를 파악하여 어플리케이션 성능 정보 수집 주기 및 정보량을 조절할 수 있어야 함
	성능추이 분석	▶ 성능 추이 분석 기능 수집된 성능 데이터/이벤트를 그래픽 형태로 분석할 수 있어야 함
		▶ 다양한 모델링 기능 What-if 모델 등의 성능 분석을 위한 다양한 모델링 기능을 제공해야 함

기능 (계속)	성능보고서 생성	▶ 성능 추이 보고서 추출기능 수집된 서버, DB, 어플리케이션, 네트워크 성능 정보를 표준으로 사용하는 Office 포맷으로 추출이 가능해야 함 (예, MS Office Excel Graph, Visio 등)
인터페이스	통합콘솔과의 이벤트 연동	▶ 통합 콘솔에 이벤트 연동 성능 임계값에 대해 발생하는 성능 이벤트를 통합 콘솔에 연동할 수 있어야 함 - UNIX서버 성능 이벤트 연동 - NT 서버 성능 이벤트 연동 - 네트워크 성능 이벤트 연동 - DB 및 미들웨어 성능 이벤트 연동 - 어플리케이션 성능 이벤트 연동
	통합 운영현황 정보 DB와 연동	▶ 통합 운영현황 정보 DB와 연동 서비스 수준 분석과 운영현황 파악을 위한 정보를 통합 운영현황정보 DB에 연동할 수 있어야 함
	용량 관리와 연동	▶ 용량 관리와 연동 어플리케이션, 서버, 데이터베이스, 네트워크 성능 분석을 통하여 용량 계획 수립에 참고할 수 있도록 수집된 성능 정보를 용량 관리 도구에 연동 할 수 있어야 함
	성능보고서 연동	▶ 성능보고서 연동 (워크플로우 기능 활용) - 워크플로우 기능을 통하여 성능 개선 보고서를 설정된 어플리케이션 담당자에게 제공할 수 있어야 함 - 워크플로우 기능을 통하여 성능 개선 보고서를 설정된 성능책임자에게 제공할 수 있어야 함

나. 구축 전략

통합적 성능관리 시스템의 최종 목표는 시스템, 데이터베이스, 네트워크 및 응용 소프트웨어 등 업무 서비스를 지원하는 모든 자원들이 최적의 상태로 운영될 수 있는 'End to End' 관리 시스템을 구축하는 것이며, 이러한 '전사적 관리 시스템(Total Enterprise Management)'의 구축을 통하여 최단 시간 내에 투자회수(ROI)의 극대화를 달성하는데 있다. 다음과 같은 기본방향에 따라 통합적 성능관리 시스템을 구축한다.

▶ 현실적인 관리체계 구축

- 관리/운영의 효율화를 위한 정책 및 가이드를 마련한다.
- 조속한 최적화 과정을 통해 투자대비 효과의 극대화를 유도한다.

▶ 통합성을 고려한 구축

- 시스템 성능만을 관리하는 단편적인 방안이 아닌, 시스템, 데이터베이스, 네트워크, 응용 소프트웨어 성능 정보를 통합 관리하는 솔루션으로 통합성을 고려하도록 구축한다. 또한 장애관리, 작업관리, 서비스수준관리, 리포트관리 등 다른 관리 분야와 통합이 용이하도록 구축한다.

▶ 미래지향적 아키텍처 구성

- 지속적인 고객요구를 충족시킬 수 있는 솔루션을 채택하여 적용한다.
- 풍부한 인력과 연구조직을 갖추고 있는 회사의 솔루션 채택한다.
- C/S 환경 외에 Intranet/Extranet 환경도 포함할 수 있는 설계 및 구현을 한다.

▶ 사전예방에 중점을 둔 관리환경 구축

- 철저한 모니터링을 통한 문제의 자동해결을 위한 절차를 적용한다.
- 축적된 데이터의 분석을 통한 사전분석 및 예측관리를 구현한다.

▶ 철저한 품질보증 및 지원

- 솔루션 제공업체의 방법론, 또는 표준 방법론을 적용한다.
- 많은 관리 시스템 구축 경험으로 검증된 인력의 기술 지원을 받는다.
- 체계적이고 수준 높은 품질보증 및 사후관리 활동을 전개한다.

▶ 수집 항목 선정 작업

- 수많은 성능 관리 항목 중 환경에 맞는 유용한 정보들만을 필터링하여 관리의 효율성을 높일 수 있도록 구축한다.

▶ 철저한 공정관리를 통한 효과적인 구축

– 공공부문의 정보시스템 센터 각 담당자와의 긴밀한 협력 및 내부적인 공정관리를 통해 비효율적인 과정의 제거, 관리 프로세스의 정립 등을 통해 완벽한 품질을 보증하도록 한다.

▶ 검증된 성능 관리의 노하우 적용

– 성능관리란 좋은 관리도구만으로 이루어지지 않기 때문에 검증된 운영방안 및 지침을 적용하여 분산 시스템을 최적의 상태로 관리한다.

– 축적된 데이터의 분석을 통한 사전분석 및 예측관리를 구현한다.

다. 구축 방법

(1) 통합적 성능관리 시스템 구축시 고려사항

▶ 일관된 시스템 관리체계 구축

: 시스템, 데이터베이스, 네트워크, 응용 소프트웨어 등 모든 정보자원에 대하여 일관된 사용자 인터페이스 기반의 관리 체계를 수립한다.

▶ 문제 해결보다는 예방에 중점을 둔 관리환경을 구축

: 철저한 작업 모니터링 및 자동 문제해결 프로시저를 개발하여 문제 자체 발생을 예방하는 시스템 환경을 구축한다.

▶ 전산자원의 가용성 극대화로 유지비용 절감

: 분산된 자원에 대하여 자원관리 및 사용량 분석을 통해 최적의 배치 및 경제적인 자원관리를 구현하며, 유지보수 비용을 절감한다.

: 복합적인 시스템 장애를 최소화하고 신속하고 정확한 원인파악 및 조치를 취한다.

▶ 확장성 있는 관리체계 구축

: 일관된 사용자 인터페이스를 비롯한 각 솔루션간 일반 서비스를 통하여 현재 도입되어 있는 관리시스템의 관리 영역을 쉽게 확장시킬 수 있어야 한다.

(2) 단계별 구축방법

통합적 성능관리 구축시 다음에 따라, 1단계부터 5단계까지 각 단계별로 구축한다.

▶ 제1단계 : 요구사항 분석(Requirement Analysis)

요구사항 및 분석은 시스템을 정의하는 프로세스이다. 문제를 정의한 후 세부적인 문제에 대한 분석을 통해 요구사항(requirements) 분석을 해야 한다. 요구사항이 기술된 문서를 명세서(specification)라고 한다. 명확한 요구사항 분석이 되지 않으면 기술자에게 사용자가 무엇을 원하는지 추측하게 만들며 잘못된 추측의 오류를 범할 수 있게 한다. 요구사항은 성능관리의 범위(scope)를 결정하며 구현과 테스트의 기준이 된다. 대형 프로젝트를 기준으로 요구사항 정의 오류는 요구사항을 결정하는 단계에서 수정하는데 비해 설계 단계에서는 5배, 구현 단계에서는 10배, 테스트 단계에서는 20배, 최종 테스트 단계에서는 50배, 그리고 유지 보수단계에서는 100배의 노력 및 비용이 필요하다는 연구 결과가 있다. 그만큼 요구사항의 정의가 중요하다.

요구 사항 및 분석은 프로젝트 관리자, 분석가, 개발자, 테스터 등 모든 관계자들에게 영향을 미치게 된다. 효과적인 요구사항 및 분석 방법은 프로젝트 위험 부담을 줄일 수 있으며, 최종 결과물이 완료될 때 까지 프로젝트를 원활하게 진행할 수 있도록 한다.

▶ 제2단계 : 설계(Conceptual/Detail Design)

성능관리 구축 및 관리팀에 속한 기술자, 관리자, 운영자, 솔루션 제공업체 기술자등 모든 사람들이 함께 협력하고, 업무 요구사항을 인지 및 공유하며, 프로세스 전반에 걸쳐 사양, 아키텍처 및 설계를 명확하게 정의하는 단계이다.

▶ 제3단계 : 커스터마이징(Customizing/Programming)

관리자 또는 사용자가 인터페이스 및 다른 요소들을 완벽하게 목적에 맞게 커스터마이징 하는 단계이다.

▶ 제4단계 : 구현(Implementation)

구현 단계는 일반적으로 총 구축 시간의 30 ~ 80% 부분을 차지하며 성능관리 시스템 구축의 중심에 위치하고 핵심이 되는 과정이다. 실제로 성능관리 시스템을 적용하는 단계이다.

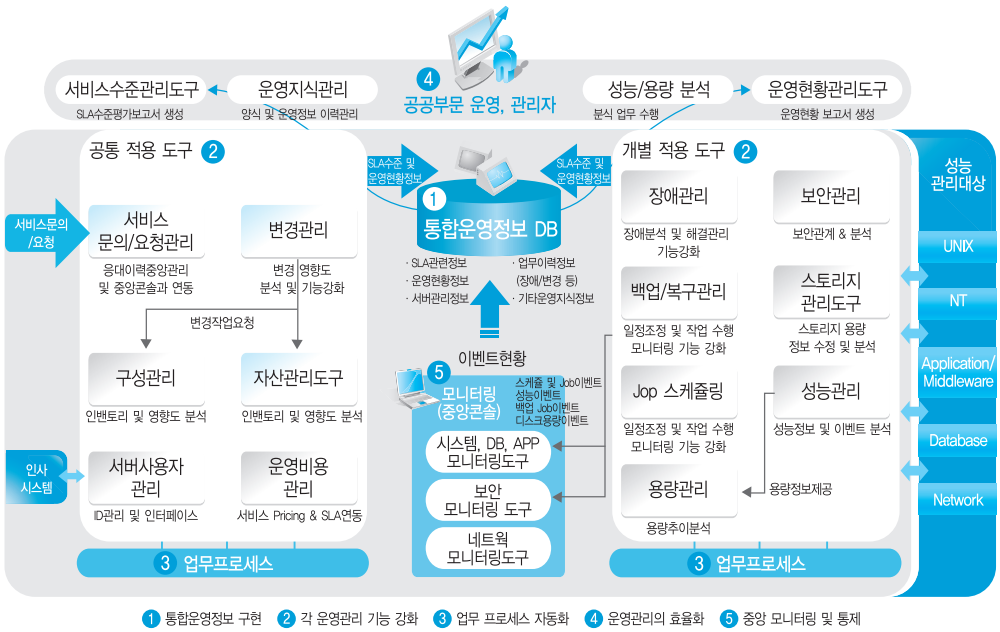
▶ 제5단계 : 운영테스트 및 사후지원(Post Implementation)

운영환경으로 전환하기 전단계로서 구현이 완료된 성능관리 시스템에 대한 실제 운영환경을 테스트하는 단계이다. 여기에서 문제가 발견 되면 해결하는 단계를 수행해야 한다.

5 3 운영방안

가. 운영 개념도

통합적 성능관리의 전체적인 운영 개념 및 흐름도는 (그림 5-1)과 같다.



(그림 5-1) 통합적 성능관리 운영 개념도

(그림 5-1)에서 보는 바와 같이 다음과 같은 5가지 프로세스별로 통합적 성능관리 및 운영관리를 수행할 수 있다.

① 통합운영 정보 구현
② 각 운영관리 기능 강화
③ 업무 프로세스 자동화
④ 운영관리의 효율화
⑤ 중앙 모니터링 및 통제

나. 고려사항

통합적 성능관리 운영시 다음의 고려사항을 참고하여 운영함으로써 효율적인 통합관리 운영이 되도록 한다.

- ▶ 성능 자료 수집 대상에 대한 분류를 명확히 한다.
- ▶ 관리 대상에 대한 명확한 분류를 한다.
- ▶ 성능 관리 대상에 대한 그룹핑 기준을 정의한다.(부문별/업무별/서버별)
- ▶ 누적치 데이터의 보관 주기를 설정한다.
- ▶ 통합 성능 관리 절차를 수립한다.
- ▶ 성능관리의 일원화를 한다.

다. 운영방안

▶ 자동 리포팅 구현

- 시스템 및 네트워크 장비들의 성능 누적치 데이터를 주기적으로 자동 취합하여 관리자에게 일/주/월간 보고 및 각 기능별 관리 서버의 독립적 운영을 보장함과 동시에 장애처리의 통합, 데이터 흐름의 중앙 집중적 관리를 함께 할 수 있도록 운영한다.
- 배치 리포팅에 의한 서비스 리포트를 주기적으로 제공하여 관리자가 시스템의 운영 현황을 올바르게 이해할 수 있게 함으로써 향후 시스템 증설, 용량 계획시에 원만한 진행을 할 수 있게 한다.
- 리포팅 생성에 소요되는 수작업 시간을 최소화한다.

▶ 성능 저하 요인 파악

- 수집된 성능 정보를 대상으로 성능 저하 요인에 대한 올바른 판단을 할 수 있도록 주기적으로 직관적인 그래프로 분석한다.
- 실시간 성능 정보에 대한 임계 치를 설정하여 성능 정보에 대한 이상 유무를 감시하며 임계치 초과 시 관리자에게 알람을 보낸다.

- 분류된 대상 시스템을 사용목적에 맞도록 그룹화하여 관리하며 이 그룹은 모든 관리기능에 적용되도록 운영한다.
- 다단계 구성이 가능하도록 운영한다.

▶ 확장성 고려

- 분류된 대상 시스템을 사용목적에 맞도록 그룹화하여 관리하며 이 그룹은 모든 관리기능에 적용되도록 운영한다.
- 다단계 구성이 가능하도록 운영한다.

부 록 성능관리 양식

□ 성능분석/조정 요청서

□ 성능분석 결과보고서

□ 성능개선 결과보고서

□ 용량증설 요청서

□ 용량증설 계획/결과보고서

성능분석/조정 요청서

년 월 일(요일)

문서번호		연계문서번호	

요청부서			처리부서		
요청부서			접수	접수번호	
담당자	(인)			접수일시	
직급/직위			성능관리담당자		(인)
전화번호			성능관리책임자		(인)
E-Mail			협조부서		(인)

☐ 성능 분석 배경

성능저하현상배경			SLA	요구수준		현상태	
성능저하업무				응답시간	처리량	응답시간	처리량
관련 프로그램							
업무 영향도							
성능 저하 발생 시점		성능분석/조정 완료시한					
관련 변경 작업 내역	*네트워크공사* *어플리케이션 패치 등*						

☐ 예상되는 성능 저하 원인

분야	서버	네트워크	DBMS	응용소프트웨어	기타
증상및측정데이터					
구성정보					
종합현황					

☐ 서비스 아키텍처 분석

서비스 아키텍처	*3 Tier환경* *C/S 환경*		
적용업무		사용시간대	
		사용빈도	
		사용자 수	

성능분석 결과보고서

년 월 일(요일)

문서번호		연계문서번호	

요청부서			처리부서			
요청부서			처리부서	접수	접수번호	
담당자	(인)				접수일시	
직급/직위				성능관리담당자		(인)
전화번호				성능관리책임자		(인)
E-Mail				협조부서		(인)

☐ 개요

분석 배경		분석 기간	
분석 목표		개선 등급	*상, 중, 하*
분석 대상	*웹서버, AP서버, DB서버*	개선 유형	*튜닝, 용량증설*

☐ 결과

구분	측정결과	분석 및 평가	비고
응답 시간	*1. 평균 응답시간* *2. 응답시간분포*		
시간당 처리량	*1. TPS 추이* *- 시간대별, 요청대상별*		
자원 사용률	*1. CPU* *2. 메모리* *3. 디스크 I/O*		
시스템 구성	*1. OS* *2. 데이터베이스* *3. 웹/WAS*		

종합평가	

성능개선 결과보고서

년 월 일(요일)

문서번호		연계문서번호	

요청부서			처리부서			
요청부서	요청부서		처리부서	접수	접수번호	
	담당자	(인)			접수일시	
	직급/직위			성능관리담당자		(인)
	전화번호			성능관리책임자		(인)
	E-Mail			협조부서		(인)

☐ 성능 개선 항목 정보

성능개선항목			SLA				
성능개선항목	성능 개선 분야	*서버, N/W, DB 응용S/W*	SLA	성능개선 전		성능개선 후	
	성능 개선 유형			응답시간	처리량	응답시간	처리량
	성능 개선 등급	*상, 중, 하*					
	관련 변경 작업 내역		성능 개선 완료 요청 일자				
	업무 영향도		성능 개선 작업 기간				

☐ 성능 개선 결과

작업 일자	작업 유형	작업 상세 내용	개선작업 결과	결과만족도	작업자
개선결과 종합					

용량증설 요청서

년 월 일(요일)

문서번호		연계문서번호	

요청부서			처리부서		
요청부서			접수	접수번호	
담당자	(인)			접수일시	
직급/직위			성능관리담당자		(인)
전화번호			성능관리책임자		(인)
E-Mail			협조부서		(인)

☐ 용량증설 사유

용량증설 사유			성능/SLA	성능개선 전		성능개선 후	
용량증설 사유	용량증설 대상	*증설요청 구성요소(CI)*		응답시간	처리량	응답시간	처리량
	관련 업무	*시스템명*					
	성능저하 발생시점						
	업무영향도		용량 증설 완료 시한				
	관련 성능 작업 내역	*관련 성능분석 및 변경 작업내역*					

☐ 성능 권고 사항

용량증설 사유		
용량증설 사유	성능 저하 증상 및 측정데이터	
	성능분석 내용 요약	
	성능저하 원인	
	권고사항	
	- 외부 전문가인 경우 소속, 성명, 연락처 등 기록	

☐ 서비스 아키텍처 분석

서비스 아키텍처			
적용업무		사용시간대	
		사용빈도	
		사용자 수	

용량증설 계획/결과보고서

년 월 일(요일)

문서번호		연계문서번호	

요청부서			처리부서			
요청부서			접수	접수번호		
담당자	(인)			접수일시		
직급/직위			성능관리담당자		(인)	
전화번호			성능관리책임자		(인)	
E-Mail			협조부서		(인)	

☐ 용량증설 계획

용량증설 대상	*증설요청 구성요소(CI)*	성능/SLA	성능개선 전		성능개선 후	
관련업무	*시스템명*		응답시간	처리량	응답시간	처리량
자원보유 여부	*현재의 보유현황 및 구입여부,방법설명*					

용량증설 사유	용량증설 계획 요약	*-주요이슈 -자원 투입 방법 -용량증설 시나리오 (증설 구현 일자/시간) -예상서비스 수준 -서비스 개선방안 및 소요비용 등*	계획수립 담당자	
			계획 작성일자	
			계획 검토결과	
			계획 검토일자	
			변경 승인일자	
			승인자	(인)

☐ 용량증설 및 결과

증설결과 요약	구현담당자	
	구현일자/시간	
	구현통보	*- 구현을 위해 사전 통보되어야 할 사항, 연관부서, 연관서비스 등*
	종료일자/시간	
	구현결과 요약	*- 구현시 문제점 - 증설 이전시점 대비 효과 - 성능 목표치 달성 여부*
	비고	

정보시스템 성능관리 지침

인쇄 : 2005년 12월
발행 : 2005년 12월

발행처 : 국무조정실 · 정보통신부

〈지침 개발 참여위원〉

· 한국전산원 이헌중 팀장
· 한국전산원 이상학 수석
· 한국전산원 문성준 선임
· 한국전산원 김은영 선임
· 한국전산원 이승한 선임
· LG CNS 김용식 과장
· 현대정보기술 김형찬 과장
· 굿어스 이진철 팀장
· 한국CA 한운교 부장

본 자료는 제22차 정보화추진위원회(2004.2.25)에 보고된 '국가기간전산망 운영실태 점검 결과'에 따른 개선대책 후속조치로 한국전산원에서 작성된 지침입니다. 본 지침과 관련된 문의는 다음을 이용하여 주시기 바랍니다.

■ **우편문의**

(우)100-170 서울특별시 중구 무교동 77번지 NCA 빌딩
한국전산원 ITA팀 정보시스템 성능관리 지침 담당자 앞

■ **전화문의 : 02-2131-0114**

- 한국전산원 ITA팀 이헌중 팀장
- 한국전산원 ITA팀 김은영 선임연구원

Guideline for Performance Management of Information System

문의처

한국전산원

100-775 서울시 중구 무교동 77번지 NCA빌딩

www.nca.or.kr Tel | 02.2131.0114 Fax | 02.2131.0409