그림으로 공부하는 IT 인프라 구조

개정판

그림으로 공부하는 IT 인프라 구조(개정판)

1쇄 발행 2020년 12월 9일

지은이 야마자키 야스시, 미나와 케이코, 아제카츠 요헤이, 사토 타카히코
감수자 오다 케이지
옮긴이 김완섭
펴낸이 장성두
펴낸곳 주식회사 제이펍

출판신고 2009년 11월 10일 제406-2009-000087호
주소 경기도 파주시 회동길 159 3층 3-B호 / **전화** 070-8201-9010 / **팩스** 02-6280-0405
홈페이지 www.jpub.kr / **원고투고** submit@jpub.kr / **독자문의** help@jpub.kr / **교재문의** textbook@jpub.kr

편집팀 김정준, 이민숙, 최병찬, 이주원 / **소통 · 기획팀** 민지환, 송찬수, 강민철, 김수연 / **회계팀** 김유미
진행 및 교정 · 교열 이주원 / **내지디자인 및 편집** 이민숙 / **표지디자인** 미디어픽스
용지 에스에이치페이퍼 / **인쇄** 한승인쇄사 / **제본** 광우제책사

ISBN 979-11-90665-20-9 (93000)
값 26,000원

그림으로 공부하는 IT 인프라 구조

개정판

야마자키 야스시, 미나와 케이코, 아제카츠 요헤이, 사토 타카히코 지음
오다 케이지 감수 / 김완섭 옮김

Jpub 제이펍

※ 드리는 말씀

- 이 책에 기재된 내용을 기반으로 한 운용 결과에 대해 저/역자, 소프트웨어 개발자 및 제공자, 제이펍 출판사는 일체의 책임을 지지 않으므로 양해 바랍니다.
- 이 책에 기재한 회사명 및 제품명은 각 회사의 상표 및 등록명입니다.
- 이 책에서는 ™, ©, ® 등의 기호를 생략하고 있습니다.
- 이 책에서 사용하고 있는 제품 버전은 독자의 학습 시점이나 환경에 따라 책의 내용과 다를 수 있습니다.
- 본문 중 일본 내의 실정에만 국한되어 있는 내용이나 그림은 일부를 삭제하거나 국내 실정에 맞도록 변경하였습니다.
- 책 맨 앞의 3계층형 시스템 구성에 관한 그림은 해당 페이지에 기재되어 있는 URL에서 PDF 버전으로도 다운로드가 가능합니다. 4~5장 학습과도 연계되어 있으니 꼭 다운로드하여 살펴보시기 바랍니다.
- 책의 내용과 관련된 문의사항은 옮긴이나 출판사로 연락주시기 바랍니다.
  - 옮긴이: jinsilto@gmail.com
  - 출판사: help@jpub.kr

차 례

옮긴이 머리말

《그림으로 공부하는 IT 인프라 구조》 초판이 한국에 소개된 것이 2015년입니다. 감사하게도 좋은 책을 번역할 수 있는 영광을 얻었고, 지금도 많은 분으로부터 사랑을 받고 있습니다. 5년이 지난 2020년 다시 새로운 모습으로 개정판을 번역할 수 있게 되어 감사하게 생각합니다. 5년 전 제가 번역했던 문장들을 비교하면서 새삼 낯설게 느껴지는 문장들도 있었지만, 다시 보니 생각보다는 나쁘지 않게 번역됐다는 사실에 안도하기도 했습니다.

5년이 지난 지금은 더 많은 것이 바뀌었습니다. 저자들의 지식도 더욱 풍부히 성장했고, 역자인 저도 번역뿐만 아니라 기술적으로도 성장한 세월이었습니다. 무엇보다 가장 많이 변한 것은 IT 관련 기술일 겁니다. 그사이 클라우드가 대세가 됐으며, 도커 등의 컨테이너가 주류로 자리 잡으면서 인프라에도 나름 많은 변화가 생겼습니다. 이런 변화에도 불구하고 이 책이 가치가 있다고 생각하는 이유는 5년이 흘러도 인프라 기술의 핵심은 변하지 않았다는 것과 이 책에서 다루고 있는 내용도 시간의 변화와 상관없이 계속 사용할 수 있는 기술들이라는 점입니다.

물론 이번 개정판도 새로운 기술(클라우드 등)이 추가되었지만, 한편으로는 생략한 내용도 있습니다. 번역을 하며 기술의 변화와는 상관없이 보다 보편적인 내용에 관점을 맞추어 독자가 실무에 활용할 수 있는 내용을 집중적으로 다루려고 하지 않았나 싶은 생각이 들었습니다.

이번에 개정판을 번역하면서 다시 한번 개정판과 이전 번역본을 대조하며 읽었는데, 정말 주옥 같은 내용이 담긴 책이라는 것을 새삼 느꼈습니다. 엔지니어 및 IT를 공부하는 많은 독자가 이 책을 접하고 천금 같은 지식을 습득하기 바랍니다.

옮긴이 **김완섭**

머리말

이 책의 초판을 집필한 이후로 5년이 지났다. IT 인프라 기술은 클라우드화를 향해 진화하고 있으며, 여러분이 사용하고 있는 시스템도 일부 또는 전부가 클라우드상에서 실행되고 있을 것이다.

클라우드를 적용하면 하드웨어 영역, 네트워크 영역, 그리고 데이터베이스나 미들웨어 영역까지 클라우드 담당 외주 업체에 맡길 수 있게 된다. 즉, 설계, 관리, 운영의 대부분을 외주 업체가 담당하게 되는 것이다. 이런 클라우드 시대에 IT 인프라에 대한 지식이 필요한 것일까?

많은 엔지니어는 필요 없다고 말할 것이다. 하지만 클라우드 자체는 미들웨어 기술이나 자동화 기술의 연장선상에 있는 것으로, 내부에서 움직이고 있는 IT 인프라가 근본적으로 바뀌는 것은 아니다. 클라우드상에서 실행되는 시스템을 구축할 때 그 내부에서 일어나고 있는 것에 관심을 가지고 IT 인프라 지식을 함께 활용하면, 더 효율적이고 구축 및 운용 비용이 적게 드는 시스템을 만들 수 있게 되리라 확신한다. 이 책을 읽어나가면서 시간이 지나도 변하지 않는 IT 인프라 기술에 흥미를 가지게 되길 바란다.

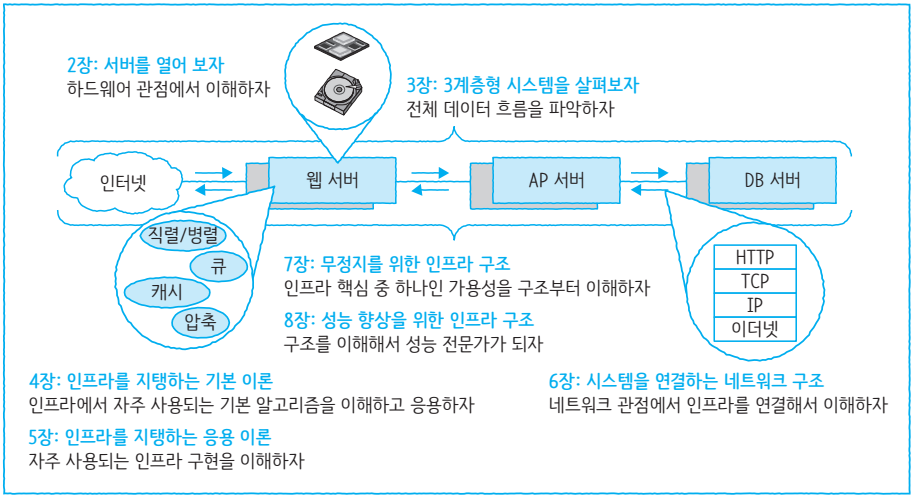
여러분이 일상에서 접하고 있는 IT 시스템은 그 구조가 매우 고도화되면서 복잡해지고 있다. 이런 복잡성을 지탱하기 위해 엔지니어의 기술 전문성이 중시되고 있으며, 결과적으로 엔지니어가 IT 시스템 전반을 살펴볼 기회가 적어지고 있다. 또, 요소 기술의 전문성이 높아짐으로써 자기 분야 외의 기술을 다룰 기회가 없어지고 있다.

하지만 현장에서는 IT 시스템 기획 및 설계 단계뿐만 아니라 성능 튜닝이나 문제 해결 등에 IT 시스템 전반을 조감하고 여러 기술을 조합해서 파악하는 능력이 요구되고 있다. IT 시스템의 세부 요소들을 보면 실제로는 평범한 상식에 의해 지탱되고 있는 것이

많다. 각 전문 영역에서 사용되고 있는 기술 요소나 이론이 실제로는 다른 영역에서도 이용되고 있다.

이 책은 평범한 상식을 기반으로 'IT 인프라'라는 영역 전체를 미시적, 거시적 관점에서 알기 쉽도록 그림을 통해 설명하고 있다. IT 인프라에 대한 상식(급소)을 이해함으로써 IT 시스템에 대한 이해가 깊어질 것이다. 또, 전문 분야 외의 새로운 기술을 접하더라도 그 본질을 이해할 수 있는 기초 체력을 단련할 수 있을 것이다.

이 책은 다음과 같이 구성돼 있다.



이 책은 IT 관련 일을 시작해서 5년차 정도까지의 엔지니어를 대상으로 하고 있다. 자신의 업무 분야에 대해서 식견이 깊어지면서 IT 인프라 전반에 대해 배우고 싶은 사람에게 추천한다.

애플리케이션 개발 엔지니어나 프로그래머도 일상에서 사용하고 있는 알고리즘이 IT 인프라와 밀접한 관련이 있다는 것을 알 수 있는 기회가 될 것이다. 또, 베테랑 엔지니어도 이전에 몰랐던 사실들을 새롭게 접할 수 있을 것이다. '예외가 있긴 하지만, 기본적으로는 이런 식으로 생각하면 이해하기 쉬워요'라는 방침을 가지고 집필했다.

감사의 글

이 책을 집필하면서 많은 분께서 도움을 주셨다. 다시 한번 감사드린다. 일부이지만 이름을 게재할 수 있도록 허락해 주신 분들의 이름을 기재해 보았다.

- 네트워크 기술 구현에 관한 검토와 조언을 주신 이토추 테크노솔루션즈의 하야시 신야 님
- 웹 서버, AP 서버 구조에 대한 검토와 조언을 주신 일본 오라클의 사쿠마 야스히로 님
- 서버 구조에 대한 검토와 조언을 주신 일본 오라클의 마루야마 타케시 님
- 커널 내부 동작에 대한 검토와 조언을 주신 이시바시 켄이치 님
- 그림을 예쁘게 만들어 주신 호리 아키코 님
- 델 EMC PowerEdge R740 서버에 관한 조언을 주신 델의 마츠다 유스케 님
- 서버 사진 촬영을 허락해 주신 일본 오라클의 야치다 노리히토 님, 오소네 아키라 님, 사와후지 코가 님

저자 소개

이 책은 일본 오라클에 재직하고 있는 컨설턴트들이 집필했다.

야마자키 야스시(山崎 泰史) — 1장, 3장, 8장 집필

오라클에서 사이트 신뢰성 엔지니어(Site Reliability Engineer)로 근무하고 있다. 셸 스크립트로 세계를 정복하려고 한다.

미나와 케이코(三縄 慶子) — 2장, 5장 일부, 7장 집필

스윔웨어 주식회사 프로페셔널 서비스 통합 본부에서 비즈니스 솔루션 기획자로 근무하고 있다. 가상화 기반 및 퍼블릭 클라우드 활용을 위한 IT 중장기 계획이나 주부로서의 감각을 살린 ROI 시뮬레이션, 시스템 아키텍처 설계 등을 담당하고 있다. 최근에는 디지털 전환(Digital Transformation) 시대의 인프라 본질에 대해 컨설팅하고 있다.

아제카츠 요헤이(畔勝 洋平) — 4장, 5장 일부, 8장 일부 집필

아마존 웹 서비스 재팬에서 빅데이터 컨설턴트로 근무 중이다. 인터넷 벤처기업에서 개발 및 인프라 운영 등 폭넓게 경험을 쌓은 후 금융 시스템 엔지니어로 일했으며, 주식회사 도완고(DWANGO)에서 DBA 및 개발팀 관리자로 일했다. 전 직장인 일본 오라클에서는 데이터베이스 컨설턴트로 금융 기관의 핵심 시스템을 설계 및 운용했다. 2017년 4월부터 현 직장인 아마존 재팬에서 근무하고 있다.

사토 타카히코(佐藤 貴彦) — 4장 일부, 5장 일부, 6장

클라우데라(Cloudera)에서 솔루션 엔지니어로 근무 중이다. 애플리케이션이 좋아서 나라첨단과학기술 대학원대학(NAIST)에서 네트워크를 전공했고, 오라클에서 배운 데이터베이스로 인프라의 매력에 빠졌다. 최근에는 데이터 자체에 대한 관심을 가지고 있다.

감수자 소개

오다 케이지(小田 圭二)

일본 오라클 주식회사에서 컨설팅서비스사업 총괄 디렉터로 근무. 관리자로 일하면서 IT 노하우 공유와 엔지니어 육성에 힘을 쏟고 있다. 주요 저서로 《그림으로 공부하는 시스템 구축을 위한 오라클 설계(絵で見てわかるシステム構築のためのOracle設計)》, 《그림으로 공부하는 오라클 구조(絵で見てわかるOracleの仕組み)》, 《그림으로 공부하는 OS/저장소/네트워크(絵で見てわかるOS／ストレージ／ネットワーク)》, 《44개 안티 패턴으로 배우는 DB 시스템(44のアンチパターンに学ぶDBシステム)》, 《신·비장의 오라클 현장 기술(新·門外不出のOracle現場ワザ)》 등이 있다.

베타리더 후기

※ 초판과 개정판의 기본 형태가 유사해 초판을 베타리딩하신 분들의 후기를 개정판에도 수록했음을 밝힙니다.

권영철(바이널아이)

회사 업무에만 빠져서 IT 인프라 구조를 막연하게만 생각했었는데, 막상 책을 보고 나니 그리 간단하지 않다는 것을 알았습니다. 이 책을 읽고 난 후 IT 인프라에 대한 이해를 넓히고 회사 업무를 해나간다면 큰 도움이 될 것 같습니다.

김민수(프리랜서)

다소 막연했던 인프라와 서버 아키텍처라는 분야를 매우 짜임새 있게 정리할 수 있는 기틀을 잡아주는 책이었습니다. 비교적 최신의 동향까지도 놓치지 않게 일러주고, 조금씩 단계를 나아갈 때마다 하나의 그림이 완성되어 가는 듯한 희열이 있더군요. 자칫 지루한 내용으로 흐를 수 있는 부분도 재미있는 예시와 유머러스한 일화로 독자의 집중도를 잃지 않게 하려는 소소한 배려도 돋보였습니다. 좋은 책을 선정하신 제이펍과 역자께도 감사의 말씀을 드립니다.

도경원(네이버)

인프라라는 거대한 숲을 모두 보기는 힘듭니다. 이 책은 거대한 숲의 나무 하나하나에 대해 자세히 설명하지는 않지만, IT 인프라를 한번 훑어보는 데는 큰 도움이 되었습니다.

송영준(줌인터넷)

자료 구조, 네트워크, 시스템 엔지니어링 등에 필요한 내용이 다양한 그림과 함께 실생활에서 찾을 수 있는 재미있는 비유로 잘 설명되어 있습니다. 다른 책에서 보였던 이해하기 힘든 설명과 지루한 구성에 힘들어했던 독자들에게 단비와 같은 책입니다.

이아름

IT 인프라 스트럭처라는 용어가 생소하던 때가 있었지만, 이제는 반드시 공부해야 하는 주제가 된 것 같습니다. 베타리딩하면서 느낀 이 책의 장점은, 글로만 보면 이해하기 어려운 부분을 그림으로 풀어내어 잘 설명한 점이라고 생각합니다.

이재빈(연세대학교)

IT 회사 입사를 준비하고 있는 분이나 종사하고 분들 모두에게 전반적인 IT 시스템 구성을 쉽게 파악할 수 있도록 도와주는 책인 것 같습니다. 더불어 생산성의 극대화를 위해서는 어떻게 인프라를 구축해야 하는지에 대한 질문에 힌트를 얻을 수도 있을 것 같습니다.

제이펍은 책에 대한 애정과 기술에 대한 열정이 뜨거운 베타리더의 도움으로
출간되는 모든 IT 전문서에 사전 검증을 시행하고 있습니다.

제 1 장

인프라 아키텍처를 살펴보자

먼저 대표적인 인프라 아키텍처를 소개한다. 역사뿐만 아니라 각각의 구조가 생겨난 이유를 생각하면서 읽도록 하자. 또한, 어떤 구조든 반드시 장점과 단점이 존재한다는 것을 이해하도록 하자.

1.1 인프라란 무엇일까?

갑작스런 질문이긴 하지만, '인프라(Infra)'라고 하면 무엇이 떠오르는가?

전기, 수도, 가스 등 가정에서 이용하는 것이나 지하철, 버스처럼 공공 목적의 인프라를 떠올릴 수 있을 것이다. 인프라를 우리말로 하면 '기반'이란 뜻으로, 여러분의 생활을 지탱하는 바탕이나 토대란 의미다. 인프라 구조 자체는 복잡하지만, 전문가에 의해 관리되고 있어서 사용자는 그 구조를 이해하지 않고도 간단히 이용할 수 있다는 특징이 있다.

'IT 인프라'도 마찬가지다. IT의 기반이 되는 것으로서 이 역시 여러분의 생활을 지탱하고 있다. 일상에서 사용하고 있는 인터넷 검색 엔진을 생각해 보자. 검색 키워드를 입력하고 검색 버튼을 누르면 많은 검색 결과를 얻을 수 있다. 이런 방대한 데이터는 어떻게 관리되고 있는지 생각해 본 적이 있는가? 이것을 지탱하고 있는 것이 IT 인프라다.

그러면 이번 장의 제목이기도 한 '인프라 아키텍처'란 무엇일까?

아키텍처란, 직역하면 '구조'라는 의미다. 여기서는 기차를 예로 들겠다. 기차에도 다양한 종류가 있지만, 그 구조 자체는 거의 같다. 전기로 움직이거나 여러 객차가 연결돼 있고, 내부에는 좌석이나 손잡이가 있다. 즉, 기차의 '구조' 또는 '아키텍처'가 확립되고, 이미 공통화돼 있다.

'인프라 아키텍처'는 IT 인프라의 '구조'를 의미한다. 인터넷 검색 시스템이나 항공 회사 티켓 발권 시스템, 편의점의 계산대 등 모두가 이용 방법이나 사용자가 다르지만 IT 인프라 위에서 동작하고 있다. 그리고 이 '인프라 아키텍처'는 실은 놀라울 정도로 닮아 있어서 거의 같은 구조를 가진 채 움직이고 있다.

이번 장에서는 현재 IT 업계에서 주류가 되고 있는 인프라 아키텍처에 대해 그림 1.1에 있는 순서대로 설명해 가도록 하겠다.

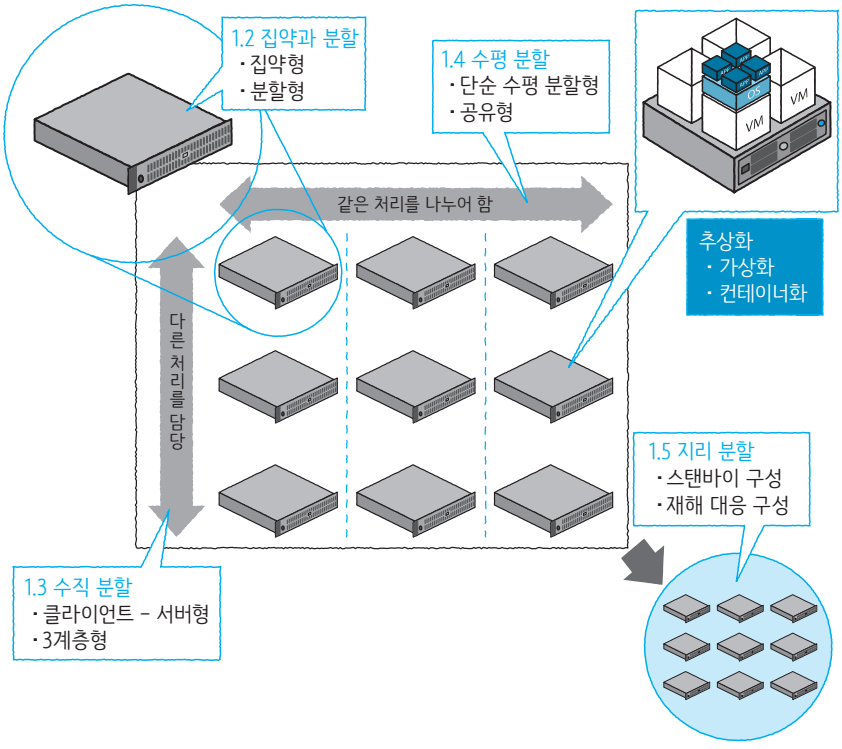


그림 1.1 이 책에서 설명하는 아키텍처

COLUMN **궁극의 아키텍처와 최적의 아키텍처는 존재하는 것일까?**

이 책에서는 다양한 아키텍처 구성과 기술 구성 요소 및 이중화, 성능 등에 대해 설명하고 있다. 읽다 보면 '어떤 시스템이든지 요건은 거의 비슷하다. 그러면 하나의 만능 아키텍처를 가지고 모든 프로젝트에 적용할 수는 없을까?', '궁극의, 최적의 아키텍처 하나만 있다면 굳이 설계를 하지 않아도 괜찮지 않을까?'라는 의문이 생길 것이다.

대답은 'NO'다. 왜냐하면 아키텍처나 설계 요소에는 반드시 장점과 단점이 공존한다. 장점만 있다면 가장 좋은 것을 취하면 되지만, 단점은 가장 영향력이 적은 것으로 선택하는 것이 어렵기 때문에 반드시 취사선택해야 할 상황이 발생한다.

그중에서도 가장 제약이 많은 것이 시스템 도입 비용이다. 예를 들어, 100만 명이 사용하는 대규모 웹 서비스의 예산과 사내에서 10명이 사용하는 시스템의 예산은 차이가 크다. 하지만 중요도라는 관점에서는 이용자에겐 양쪽 시스템이 다 중요하다. 많은 현장에서 비용 제약이 심해지고 있는 가운데, 시스템의 가장 중요한 장점은 살리고 단점을 최소화하도록 설계하는 것이 중요하다.

'설계? 다 똑같은 거 아닌가?'라는 얘기를 들어도 개의치 말고, 시스템에 따라 중요한 사항을 잘 찾아서 최적의 설계를 하도록 하자.

1.2 집약형과 분할형 아키텍처

IT 인프라는 컴퓨터로 구성된다. 기본적인 구성 방식에는 '집약형'과 '분할형'이 있다. 각각의 장단점을 비교해 보도록 하자.

1.2.1 집약형 아키텍처

옛날 영화에 등장하는 컴퓨터 장면에서는 방 하나에 가득 차는 기계와 거대한 카세트 테이프 같은 장치, 청색, 녹색, 황색 등으로 깜빡이는 램프, 그리고 안경을 낀 과학자들이 등장한다. 기업에서 사용하는 컴퓨터는 이 정도로 거대하진 않지만, IT 시스템의 여명기에는 대형 컴퓨터를 이용해서 모든 업무를 처리하는 형태가 대부분이었다(그림 1.2).

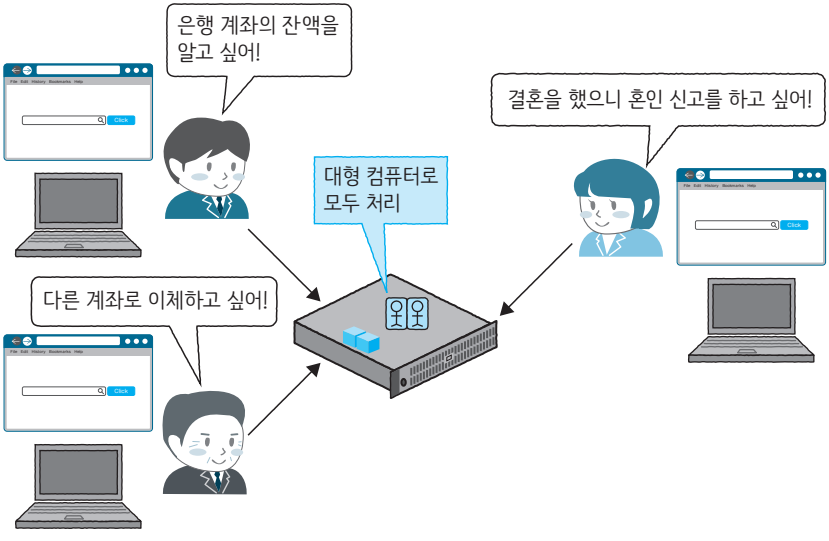


그림 1.2 집약형 아키텍처

이런 대형 컴퓨터는 '범용 장비', '호스트', '메인 프레임' 등으로 불렸다. 시스템 아키텍처라는 관점에서는 하나의 컴퓨터로 모든 처리를 하기 때문에 '집약형'이라고 할 수 있다. 집약형의 최대 장점은 구성이 간단하다는 것이다.

집약형 아키텍처에서는 해당 기업의 주요 업무를 모두 한 대로 처리하기 때문에 장비 고장 등으로 업무가 멈추지 않도록 여러 고민을 하고 있다. 예를 들어, 그림 1.3에 있는 것처럼 컴퓨터를 구성하는 주요 부품은 모두 다중화돼 있어서 하나가 고장 나더라도 업무를 계속할 수 있다. CPU를 포함한 구성 요소에 대해서는 2장에서 자세히 다루도록 한다.

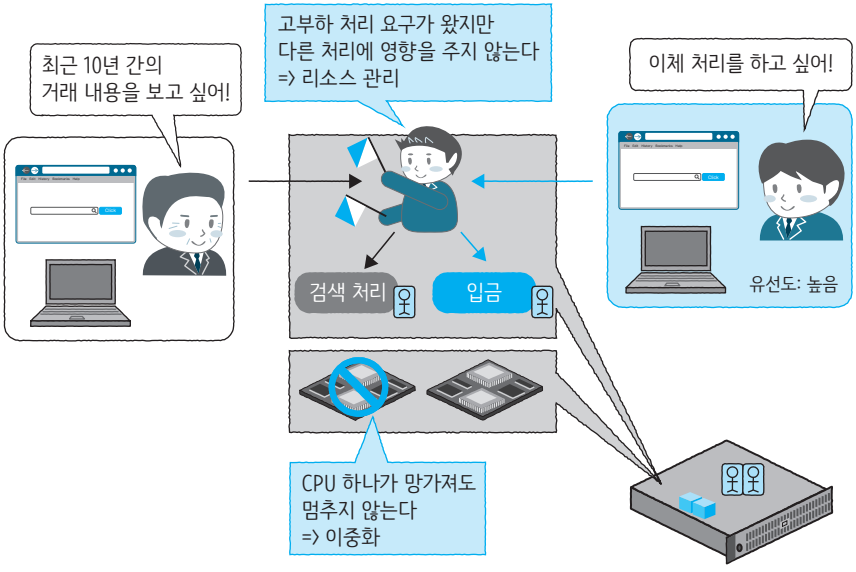


그림 1.3 대형 컴퓨터에 대한 고민

또, 복수의 서로 다른 업무 처리를 동시에 실행할 수 있도록 유한 리소스 관리를 한다. 이를 통해 하나의 처리가 실수로 대량의 요청을 보내더라도 다른 처리에 영향을 주지 않도록 되어 있다. 한 대의 컴퓨터라고 하지만, 그 안에 마치 여러 사람이 동거하고 있는 모습이라 할 수 있다.

많은 기업에서 아직까지 사용되고 있으며, 주로 '기간 시스템'이라 불리는 기업 내 핵심 업무 시스템에서 이용하고 있는 경우가 많다. 예를 들어, 은행이라면 '계정 시스템'이 여기에 해당된다.

단, 대형 컴퓨터는 도입 비용 및 유지 비용이 큰 경향이 있다. 또, 대형 컴퓨터의 파워가 부족하면 다른 한 대를 별도로 구매해야 해서 비용이 매우 많이 들며, 확장성에도 한계가 존재한다는 단점이 있다. 현재는 가격이 싸고 확장성이 높은 분할형이 주로 사용되고 있다. 이 구조에 대해서는 다음 절에서 설명하겠다.

장점

- 한 대의 대형 컴퓨터만 있으면 되므로 구성이 간단하다
- 대형 컴퓨터의 리소스 관리나 이중화에 의해 안정성이 높고 고성능이다

단점

- 대형 컴퓨터의 도입 비용과 유지 비용이 비싸다
- 확장성에 한계가 있다

1.2.2 분할형 아키텍처

분할형 아키텍처는 그림 1.4처럼 여러 대의 컴퓨터를 조합해서 하나의 시스템을 구축하는 구조다.

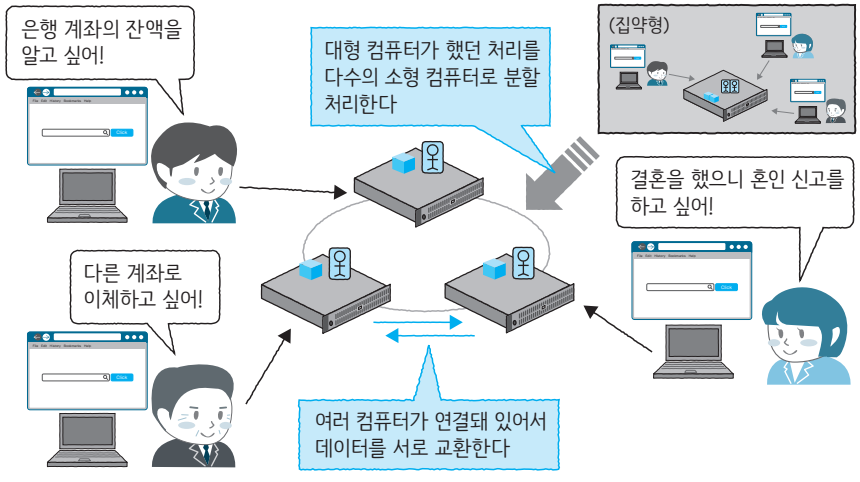


그림 1.4 분할형 아키텍처

제2장

서버를 열어 보자

이번 장에서는 하드웨어 장비를 소개하고 그 내부에서 데이터가 어떻게 흐르고 있는지 설명한다.

2.1 물리 서버

2.1.1 서버 외관과 설치 장소

앞 장에서는 대표적인 시스템 아키텍처를 소개했다. 아키텍처 전체를 구상할 때는 먼저 서버라는 단위로 생각한다는 것을 이해했을 것이다. 이번 장에서는 물리 서버 내부 구조에 대해 더 자세히 살펴보도록 한다.

여러분은 데이터 센터나 서버실에 들어가 본 적이 있는가? 서버가 대량으로 설치돼 있고, 서버에서 나오는 열기를 식히기 위해 실내 온도를 항상 낮게 설정할뿐더러 빛도 들어오지 않는 환경이다. 서버에겐 친절하지만 사람에겐 그렇지 못한 장소로, 그림 2.1과 같다.

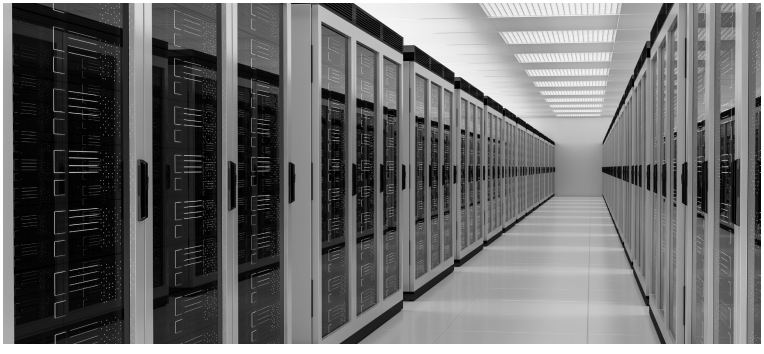


그림 2.1 서버실 예 출처 sdecoret/Shutterstock.com

서버는 랙(Rack)이라는 것에 장착된다. 랙에는 서버 외에도 HDD가 가득 장착돼 있는 저장소나 인터넷 및 LAN을 연결하기 위한 네트워크 스위치 등도 탑재돼 있다(그림 2.2).

어째서 랙에 서버가 딱 맞게 설치되는지 신기하게 생각한 적이 없는가? 사실은 서버 랙에도 규격이 있는데, 대부분의 랙은 폭이 19인치다. 또, 높이는 한 칸에 약 4.5cm로 40~46개 칸으로 이루어져 있다. 이 한 칸을 1U라고 하며, 서버 높이는 이 단위를 따르고 있다. 이 때문에 2U 서버는 2칸(높이 약 9cm)의 서버임을 의미한다[1].

1 옮긴이 최근에 진행한 프로젝트에서는 대량의 서버가 필요했다. 하지만 데이터 센터가 거의 다 찬 상태라 비어 있는 랙이 드물었다. 처음 고려했던 것이 2U 서버, 즉 랙 2칸을 차지하는 서버였지만, 조금이라도 공간을 확보하기 위해서 사양을 1U 서버로 변경한 경험이 있다. 1U 서버는 물론 크기가 작지만, 그만큼 성능이 줄어드는 측면이 있으므로 잘 고려하여 서버를 선택하자.

전원이나 네트워크 케이블 배선 등은 모두 랙 뒷면에서 연결된다(그림 2.3).

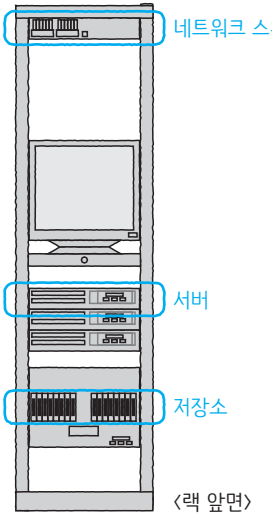


그림 2.2 랙 앞면의 모습

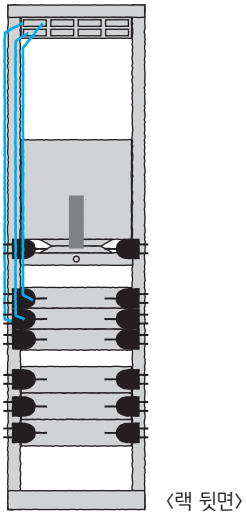


그림 2.3 랙 뒷면의 모습

서버 설치 시에 중요한 정보는 다음과 같다.

- 서버 크기(U)
- 소비 전력(A)
- 중량(Kg)

다음은 대표적인 서버 아키텍처 중 하나인 인텔의 CPU를 사용한 IA 서버에 대한 설명이다. 먼저, 서버 사진을 보자(그림 2.4).

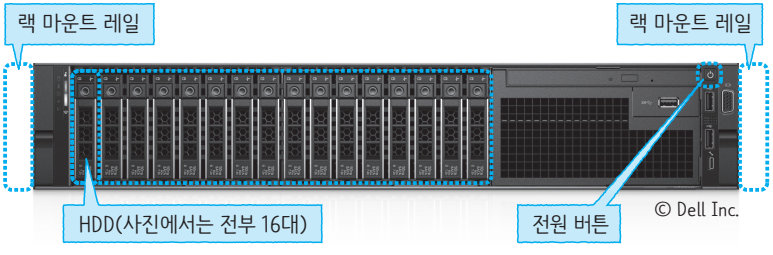


그림 2.4 서버 전면(델 EMC PowerEdge R740)

이것은 델 테크놀러지(Dell Technologies)가 판매하는 델 EMC PowerEdge R740(이하 PowerEdge 740)이라는 모델의 서버 전면 사진이다. 일반적인 서버는 이렇게 옆으로 긴 형태다. 또한, 옆에 랙 마운트 레일(Rack Mount Rail)이라는 것이 있어서 장롱 서랍처럼 설치할 수 있다.

전면에는 HDD나 전원 버튼 등이 있다. HDD는 교체하기 쉽게 되어 있어서 손으로 당겨 꺼낼 수 있다.

2.1.2 서버 내부 구성

이런 서버는 위쪽 뚜껑을 열 수 있다. 그림 2.5는 뚜껑을 열어 놓은 사진이다.

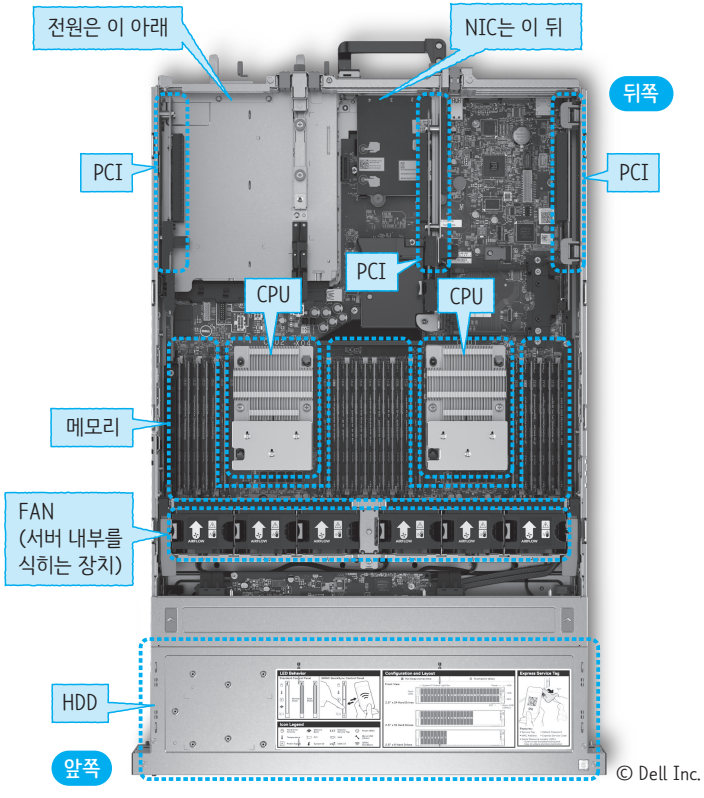


그림 2.5 서버 내부(델 EMC PowerEdge R740). 컴포넌트는 PC와 동일

PC 부품과 같은 종류가 들어 있는 것을 알 수 있다.

각 부품이 어떻게 연결돼 있는지 그림으로 표현하면 다음과 같다(그림 2.6). 이것은 PowerEdge R740의 CPU인 Intel Xeon 프로세서를 사용한 버스 접속의 일반적인 예다.

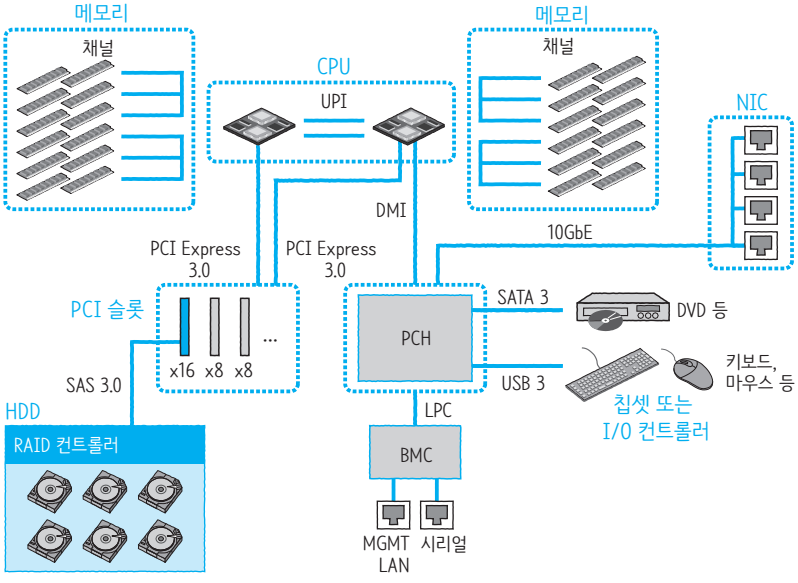


그림 2.6 컴포넌트들은 버스(Bus)로 연결된다

CPU, 메모리, HDD 등의 위치 관계를 알 수 있을 것이다. 컴포넌트를 연결하는 선을 '버스(Bus)'라고 한다.

그림 2.6 왼쪽 상단을 보자. CPU가 두 개 연결돼 있고, 그 옆에 메모리가 배치돼 있다. CPU와 메모리는 물리적으로 직접 연결되는 것을 알 수 있다.

왼쪽 하단에는 PCI Express 슬롯이라는 것이 있다. 이것은 외부 장치를 연결하는 곳이다. Xeon 확장 프로세서 아키텍처에선 CPU가 PCI를 직접 제어한다.

계속해서 오른쪽 상단을 보자. 이 서버에서는 칩셋이 네트워크 인터페이스를 4개까지 직접 제어할 수 있다. CPU를 중심으로 생각하면 USB 네트워크 인터페이스는 메모리에 비해 멀리 있다는 것을 알 수 있다. 이 이유에 대해서는 뒤에서 설명하겠다.

아래에는 BMC(Baseboard Management Controller)라는 컴포넌트가 있다. 이것은 서버의 H/W 상태를 감시하며, 독립적으로 움직인다. 예를 들어 서버의 H/W에서 장애가 발생한 경우, BMC 콘솔을 통해 서버 상태를 확인하거나 네트워크로 접속해서 서버를 원격으로 재시작할 수 있다.

서버 내에는 이외에도 다수의 컴포넌트가 존재하지만, 이 책에서는 주요 등장 인물인 CPU, 메모리, HDD, 네트워크 인터페이스, 버스를 중심으로 설명한다.

이미 눈치 챈 사람도 있겠지만, 서버와 PC는 물리적으로는 기본 구성이 같다. 전원이 이중화돼 있어서 장애에 강하거나 대용량 CPU나 메모리가 탑재돼 있는 정도가 PC와 다른 점이다. 따라서 각 하드웨어 제조사가 경쟁하는 것은 CPU나 메모리 등의 컴포넌트 자체를 개선하거나 컴포넌트 배치나 버스 구성, 하드웨어를 가동시키기 위한 펌웨어를 개선해서 성능을 차별화하는 것이다.

이후로는 서버에 어떤 기술이 반영돼 있는지와 이 기술을 통해 서버를 어떻게 물리 설계하면 좋을지를 설명하겠다. 하드웨어의 진화가 매우 빠르기 때문에 여기서 소개하는 기술, 대역 속도 등은 곧 바뀔 수도 있다. 여기서는 어떤 개념이나 기술 방식이 존재하는지만 기억해 두기를 바란다.

2.2 CPU

CPU는 Central Processing Unit의 약자다. 서버 중심에 위치해서 연산 처리를 실시한다. 그림 2.7은 PowerEdge R740에 탑재되어 있는 Xeon 확장형 프로세서로, CPU의 종류 중 하나다.

© Intel Corporation

표면. 대량의 전기 신호를 처리하기 때문에 발열이 심하다. 이 부분에는 보통 '냉각기'가 설치된다

뒷면. 실제로는 시스템 포트에 장착돼 있어서 보이지 않는다. 주변을 감싸고 있는 대량의 핀이 버스(뒤에서 설명)에 연결돼 있어서 메모리나 디스크와 데이터를 교환한다

그림 2.7 CPU(Xeon 확장형 프로세서, Intel Xeon Gold 5115)

제 8 장

성능 향상을 위한 인프라 구조

인프라 아키텍처를 그림으로 그려 보면 병목 현상이 발생하기 쉬운 위치를 파악해서 개선 방향을 검토할 수 있다. 이번 장에서는 병목 현상 개념과 다양한 병목 현상 사례를 소개하니 핵심을 잘 이해해서 자기 것으로 만들도록 하자.

8.1 응답과 처리량

8.1.1 성능 문제의 두 가지 원인

시스템 성능 문제는 대부분 사용자 불만을 통해 인지하게 된다.

- '시스템이 느려서 사용할 수 없어'
- '클릭한 후 아무리 기다려도 화면이 뜨지 않아'
- '일괄 처리가 아침이 되도 끝나지 않아'
- '오후에만 시스템이 갑자기 느려져'

표현은 모두 다르지만 같은 얘기를 하고 있다. '사양이라서 어쩔 수 없어요'라고 대답할 수 없는 것이 인프라 기술자의 운명이다. 그러면 이런 불만이 있을 때 인프라 관점에서는 다음 사항을 사용자로부터 확인해야 한다.

- '응답 속도가 느립니까?'
- '처리량이 낮습니까?
- '아니면 둘 다입니까?'

시스템 성능을 가리킬 때 응답(Response)과 처리량(Throughput)이라는 지표가 자주 사용된다. 응답은 처리 하나당 소요 시간을 의미하며, 처리량은 단위 시간당 처리하는 양을 의미한다. 일상에서도 사용하는 용어이지만, 이 두 가지를 혼동해서 사용하는 경우가 많아서 명확히 구별하는 것이 중요하다[1].

예를 들어, 검색 엔진에서 키워드를 입력해서 '검색' 버튼을 누른 후 검색 결과가 표시되기까지 걸리는 시간이 응답 시간(Response Time) 이다. 응답에 걸리는 시간으로, 명칭 그대로다. 한편 처리량은 이 검색 엔진이 초당 받아 들이는 사용자 수에 해당한다. 응답은 '서비스를 이용하는 한 명의 사용자' 관점 지표인 반면, 처리량은 '서비스 제공자' 관점의 지표라고 할 수 있다.

1 턴 어라운드 타임(Turn Around Time)이라는 용어도 있다. 응답은 응답이 돌아오기까지 걸리는 시간이고, 턴 어라운드 타임은 처리가 완료되기까지 걸리는 시간을 가리킨다. 이 책에서는 응답으로 통일한다.

그림 8.1은 편의점을 예로 들고 있는데, 조금 더 자세히 설명하겠다. 이 가게에는 계산대가 두 곳에 있으며, 계산 담당자 두 명의 능력이 동일하다고 가정한다. 그리고 고객이 두 계산대에 균등하게 줄을 서 있다고 하자. 이때 줄의 끝에 선 후 계산이 끝나기까지 걸리는 시간이 응답 시간이다. 또한, 한 대의 계산대에서 한 명당 1분이 걸린다고 하자. 계산대가 두 개이기 때문에 1분당 두 명분을 정산할 수 있다. 이것을 처리량이라고 한다.

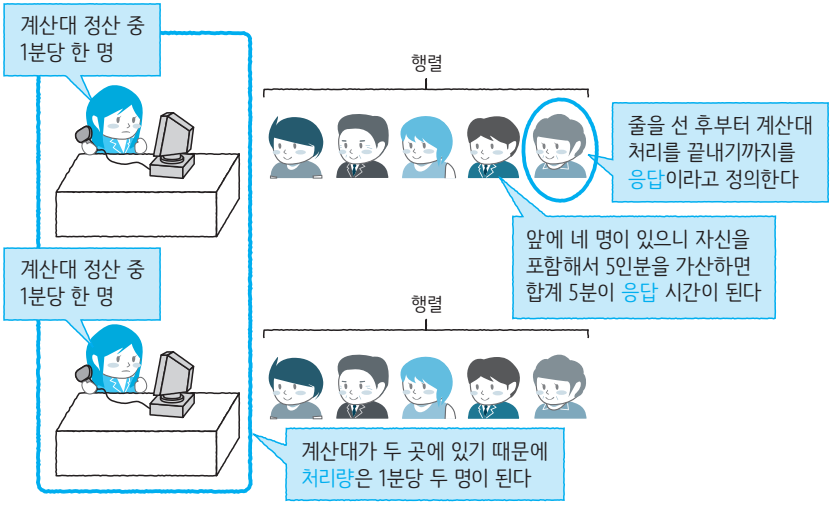


그림 8.1 편의점 계산대를 예로 든 처리량과 응답

이어서 그림 8.2는 편의점 예를 시스템 구성도로 표현한 것이다. 웹/AP/DB 서버를 한 세트로 하는 시스템이 시스템 전체에서 1분간 10000HTTP 요청을 처리하는 처리량을 가지고 있다고 하자. 응답 시간은 한 명의 사용자가 본 지표이기 때문에 사용자가 웹 브라우저에서 특정 조작을 한 후 눈에 보이는 결과가 반환되기까지의 시간이다. 이것은 웹 서버의 응답뿐만 아니라 AP 서버나 DB 서버의 응답도 포함한다는 것에 주의하자.

실제 시스템에서는 단일 사용자 응답 시간만으로는 부족하기 때문에 여러 사용자의 평균값을 이용한다. 이때 통계학에서 사용되는 '퍼센타일(Percentile)'[2] 개념을 이용한다. 극단적으로 응답 시간이 긴 사용자는 다른 문제를 내포하고 있을 수 있기 때문에 오차

2 퍼센타일은 값 분포를 고려해서 그룹화한 것을 기준으로 평균 등을 구하는 계산 방식이다. 이것을 이용하면, 예를 들어 '100명 중 90명이 본 응답 시간이 1초이고 나머지 10명은 5초였다'와 같이 사용자 관점에 가까운 평균값을 구할 수 있다.

라고 생각하고 평균값에 포함하지 않는다. 10%의 사용자 값을 버리고 나머지 90% 사용자의 평균 응답 시간을 이용하는 형태다.

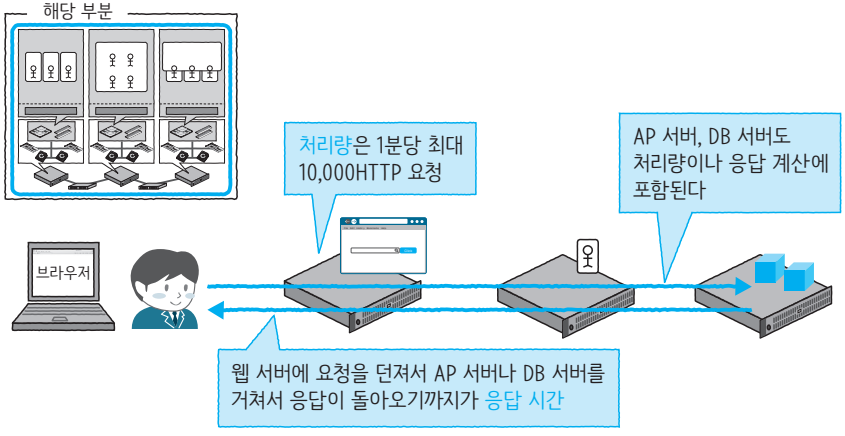


그림 8.2 3계층형 시스템 관점의 처리량과 응답

COLUMN 가장 중요한 응답 시간은?

시스템 관점에서는 개별 사용자 응답보다는 앞서 설명한 퍼센타일 값을 중시한다. 예를 들어, 5명만 높은 응답 시간을 유지하고 나머지 사용자는 응답에 만족하지 못하는 상황이라면 의미가 없기 때문이다. 반대로, 응답 속도 저하를 느끼고 있는 것이 극히 일부 사용자라면 시스템이라기보다는 사용자 PC에 문제가 있을 가능성이 높다.

프로젝트의 성능 테스트 단계에서는 이것으로 충분하지만, 대부분의 현장에서는 '회사의 높은 사람이 실제로 시스템을 사용해 보는' 경우가 발생하기도 한다. 성능 테스트에서 결과가 좋더라도 회사의 높은 사람이 봤을 때 느린 시스템은 결국 사용자에게 배포될 수 없다. 게다가 회사의 높은 사람이 보는 것은 대부분 응답 시간 측면이고 처리량은 무시되기 때문에 주의가 필요하다.

또한, 이런 분들은 가끔 '모든 화면이 3초 이내에 표시되도록 해!'라고 선언하기도 한다. 이것은 데이터 양이나 서버 성능을 무시한 지시다. 이럴 때는 다음과 같은 예로 대응해야 한다. '현재 구성은 쿼리 단위로 총 10억 건의 명세서를 사용해서 집계하고 있습니다. 해당 목표치를 달성하는 것은 비현실적이기 때문에 표시 내용을 확 줄여서 요약 형태로 만들어야 합니다. 그렇게 할까요?'라고 하면 상대방도 '흠, 그럼 어쩔 수 없겠군'이라며 이해할 것이다.

이런 문제도 모든 것을 메모리에 저장하는 시대가 오면 해결될 거라고 말하는 사람도 있지만, 나는 시스템 성능 진화와 데이터 양 증가는 같은 속도로 진행되기 때문에 아무리 시간이 지나도 성능 문제는 사라지지 않을 것이라고 생각한다.

8.1.2 응답 문제

그림 8.3에서는 응답 시간에 포함되는 시간을 도식화하고 있다. 사용자 체감 시간에는 각 계층의 처리 시간이 포함되므로 응답 문제가 발생하는 위치는 로그나 실제 장비 시험 등을 통해서 구체적으로 어떤 계층에서 응답 지연이 발생하고 있는지 파악해야 한다. 시스템에 문제가 있다는 사용자 불만을 접수해서 확인해 보면, 사용자가 이용하던 웹 브라우저의 처리 속도(렌더링 속도)가 느려서 문제가 발생하는 경우가 있다. 농담처럼 들리지만 실제로 자주 있는 일이다.

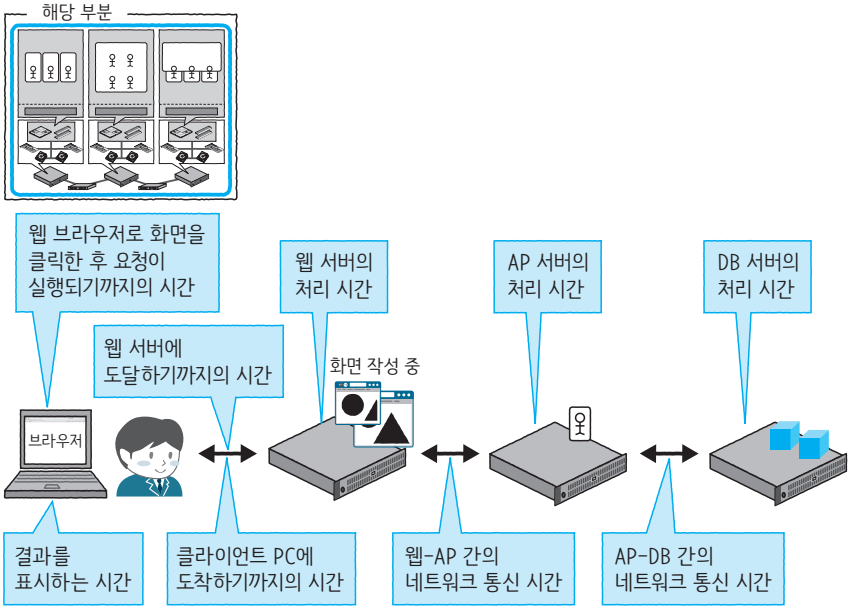


그림 8.3 응답 시간에 포함되는 시간

각 서버의 응답 시간에 대해서는 로그 등을 보면 어느 정도 문제 파악이 가능하다. 그러면 네트워크 문제는 어떨까?

응답의 중요한 요소로 시스템에 도달하기까지의 시간과 돌아오기까지의 시간이 있다. 전기 회선을 통과하기 때문에 이 시간이 빛의 속도와 같다고 생각할 수도 있다. 그렇다면 1초에 지구를 7바퀴 반 돌 수 있는 것일까? 아니다. 예를 들어, 웹 브라우저가 액세

스하는 경우를 생각해 보자. 요청이 다양한 스위치나 라우터를 경유해서 최종 시스템에 도달하게 된다. 개별 스위치를 통과하는 데 걸리는 시간은 제로가 아니다. 경로가 복잡할수록 지연이 커진다. 이것은 정보 교환이 단방향이 아니라서 반드시 어떠한 형태든 응답을 해 가면서 진행을 해야 하기 때문이다.

그림 8.4에서 보듯이 인터넷 경유이든 사내 시스템이든 정도의 차이는 있지만 비슷한 지연이 발생한다. 기회가 있으면 꼭 데이터 센터에서 직접 시스템에 접속해 보자. 속도가 다르다는 것을 체감할 수 있을 것이다.

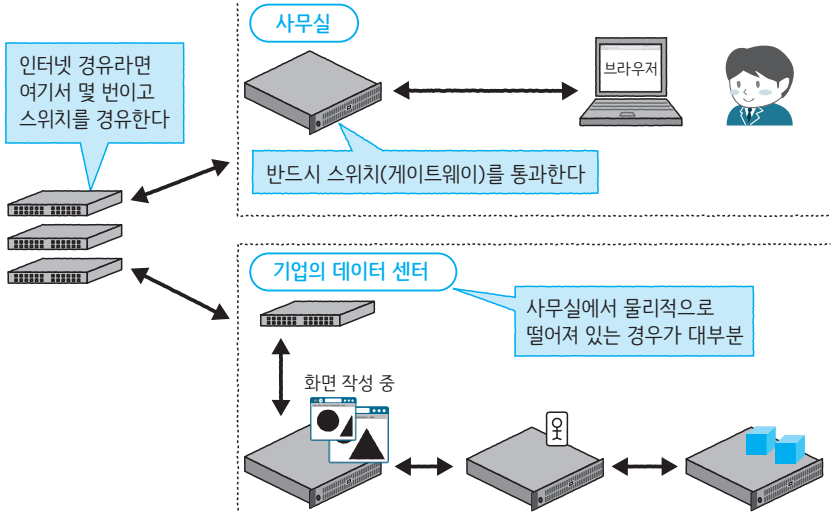


그림 8.4 시스템에 도달하기까지의 경로

다시 이번 장 서두에서 했던 질문으로 돌아가 보자. '시스템이 느려', '클릭해도 화면이 표시되지 않아'와 같은 문제는 응답 시간에 문제가 있을 가능성이 높기 때문에 먼저 이 관점으로 조사해야 한다.

앞서 빛이 1초당 지구를 7바퀴 반 돈다는 얘기를 했었지만, 모든 응답 시간에는 반드시 물리적 제약이 존재한다. 데이터라는 정보가 물리적으로 이동하는 이상은 어쩔 수 없는 제약이다. 이것을 개선하기 위한 구조로 4장에서 소개한 데이터 구조나 탐색 알고리즘 등을 적용하기도 하지만, 이것도 역시 한계가 있다.