QUESTIONS FROM STUDENTS

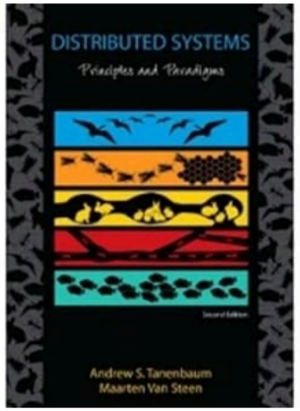
– (1) What does the exam look like?

QUESTIONS FROM STUDENTS

- (1) What does the exam look like?
  - midterm: five questions, final: eight questions (tentative)
  - question 1: answer true or false for a list of statements
  - question 2-4:
    - write a MapReduce program for [..] in pseudocode
    - given a diagram, explain if it satisfies [..] and explain why
  - question 5: project-related

QUESTIONS FROM STUDENTS

– (2) Textbooks (or fun reading)?



Distributed Systems: Principles and Paradigms (2nd Edition) 2nd Edition

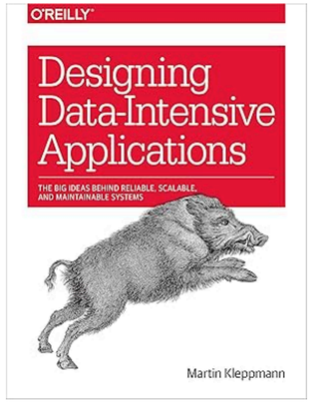
by Andrew S. Tanenbaum (Author), Maarten Van Steen (Author)

4.3 ★★★★½ (67) 4.0 on Goodreads 474 ratings See all formats and editions

Virtually every computing system today is part of a distributed system. Programmers, developers, and engineers need to understand the underlying principles and paradigms as well as the real-world application of those principles. Now, internationally renowned expert Andrew S. Tanenbaum – with colleague Martin van Steen – presents a complete introduction that identifies the seven key principles of distributed systems, with extensive examples of each. Adds a completely new chapter on architecture to address the principle of organizing distributed systems. Provides extensive new material on peer-to-peer systems, grid computing and Web services, virtualization, and application-level multicasting. Updates material on clock synchronization, data-centric consistency, object-based distributed systems, and file systems and Web systems coordination. For all developers, software engineers, and architects who need an in-depth understanding of distributed systems.

Report an issue with this product or seller

Books › Computers & Technology › Databases & Big Data › Data Modeling & Design



Read sample ▶ Preview

Follow the author

Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems 1st Edition

by Martin Kleppmann (Author)

4.7 ★★★★½ (5,287) 4.7 on Goodreads 10,076 ratings

#1 Best Seller in MySQL Guides See all formats and editions

Save 5% on 3 select item(s) Shop items ›

Data is at the center of many challenges in system design today. Difficult issues need to be figured out, such as scalability, consistency, reliability, efficiency, and maintainability. In addition, we have an overwhelming variety of tools, including relational databases, NoSQL datastores, stream or batch processors, and message brokers. What are the right choices for your application? How do you make sense of all these buzzwords?

In this practical and comprehensive guide, author Martin Kleppmann helps you navigate this diverse landscape by examining the pros and cons of various technologies for processing and storing data. Software keeps changing, but the fundamental principles remain the same. With this book, software engineers and architects will learn how to apply those ideas in practice, and how to make full use of data in modern applications.

- Peer under the hood of the systems you already use, and learn how to use and operate them more effectively
- Make informed decisions by identifying the strengths and weaknesses of different tools
- Navigate the trade-offs around consistency, scalability, fault tolerance, and complexity
- Understand the distributed systems research upon which modern databases are built

Read more

Report an issue with this product or seller

FOUR FEATURES NEW IN TODAY'S CLOUDS

- (1) Massive scale.
- (2) On-demand access: Pay-as-you-go, no upfront commitment.
- (3) Data-intensive Nature: What was MBs has now become TBs, PBs and XBs.
- (4) New Cloud Programming Paradigms

(1) MASSIVE SCALE

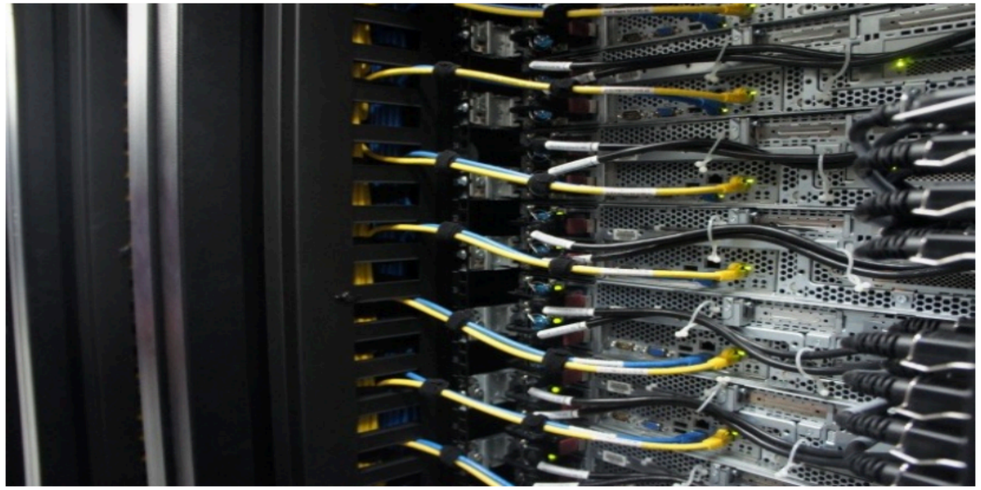
- Facebook [GigaOm, 2012]: 30K in 2009 -> 60K in 2010 -> 180K in 2012
- Microsoft [NYTimes, 2008]
  - 150K machines. Growth rate of 10K per month
  - 80K total running Bing
  - In 2013, Microsoft Cosmos had 110K machines (4 sites)
- Yahoo! [2009]:
  - – 100K, Split into clusters of 4000
- AWS EC2 [Randy Bias, 2009]
  - 40K machines – 8 cores/machine
- eBay [2012]: 50K machines
- HP [2012]: 380K in 180 DCs
- Google [2011, Data Center Knowledge] : 900K

Q: any guess on how many servers Microsoft Azure have today?

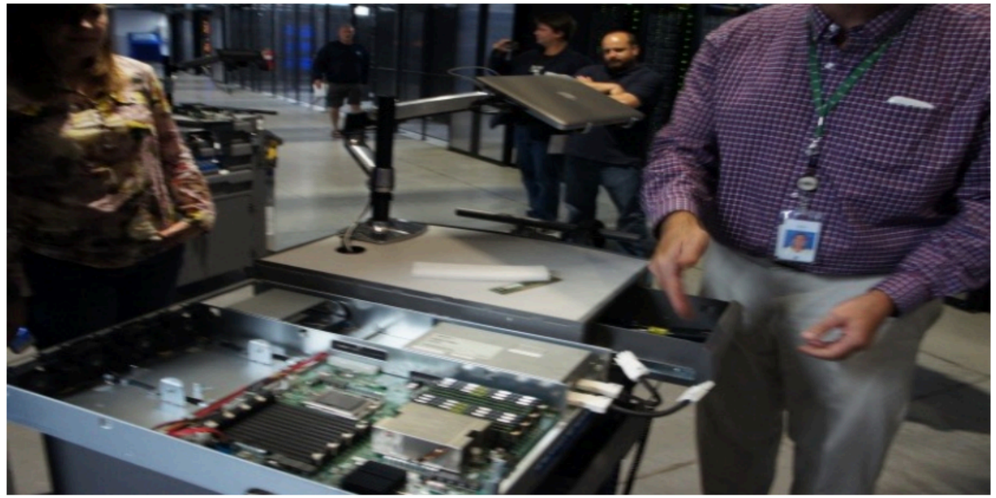
What does a datacenter look like from inside?

Servers

Front



Back



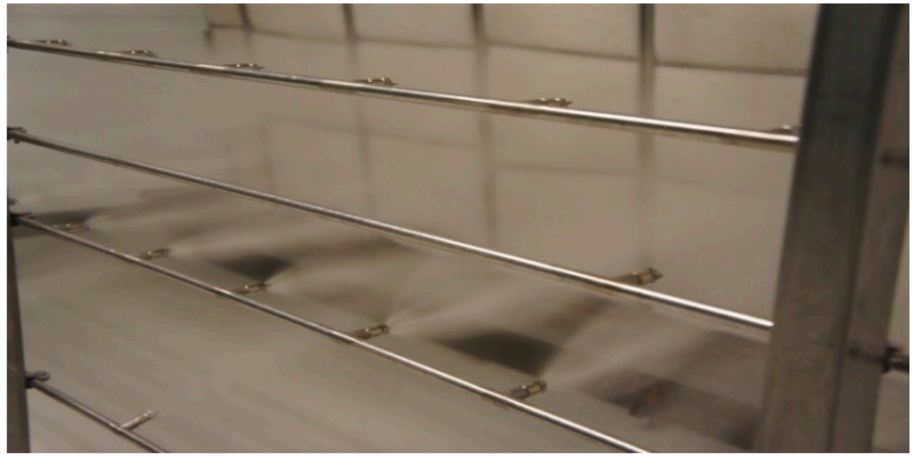
In

Some highly secure (e.g., financial info)

Cooling

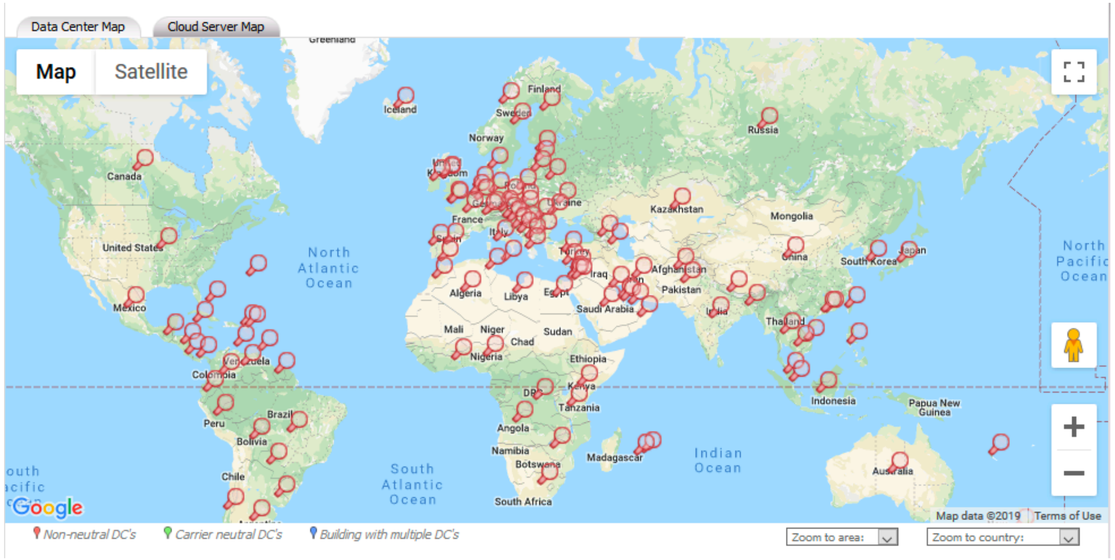
Air sucked in from top (also, Bugzappers)

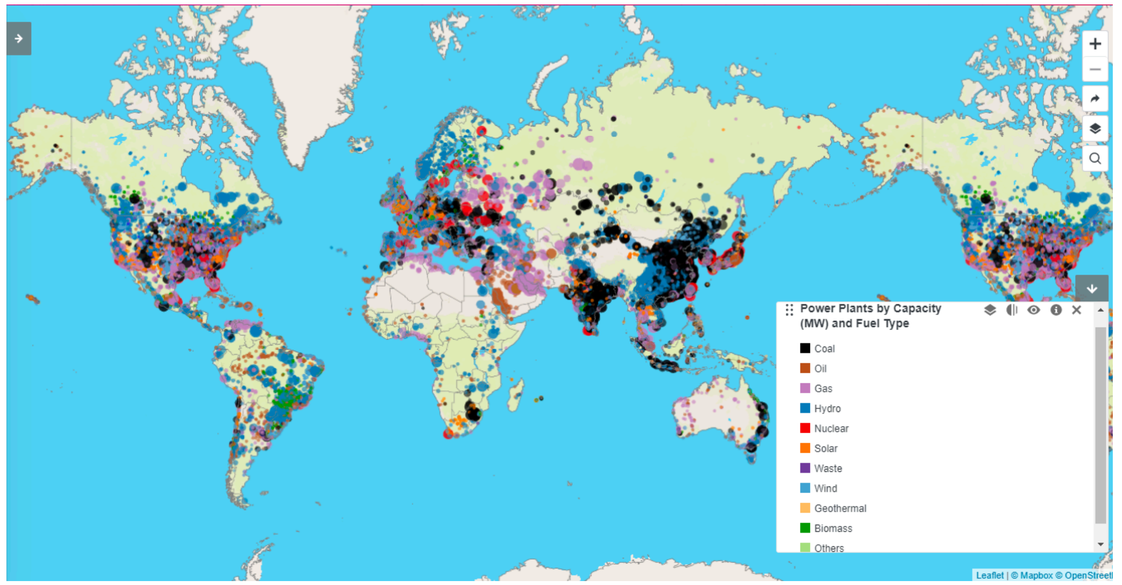
Water purified



Water sprayed into air

15 motors per server bank

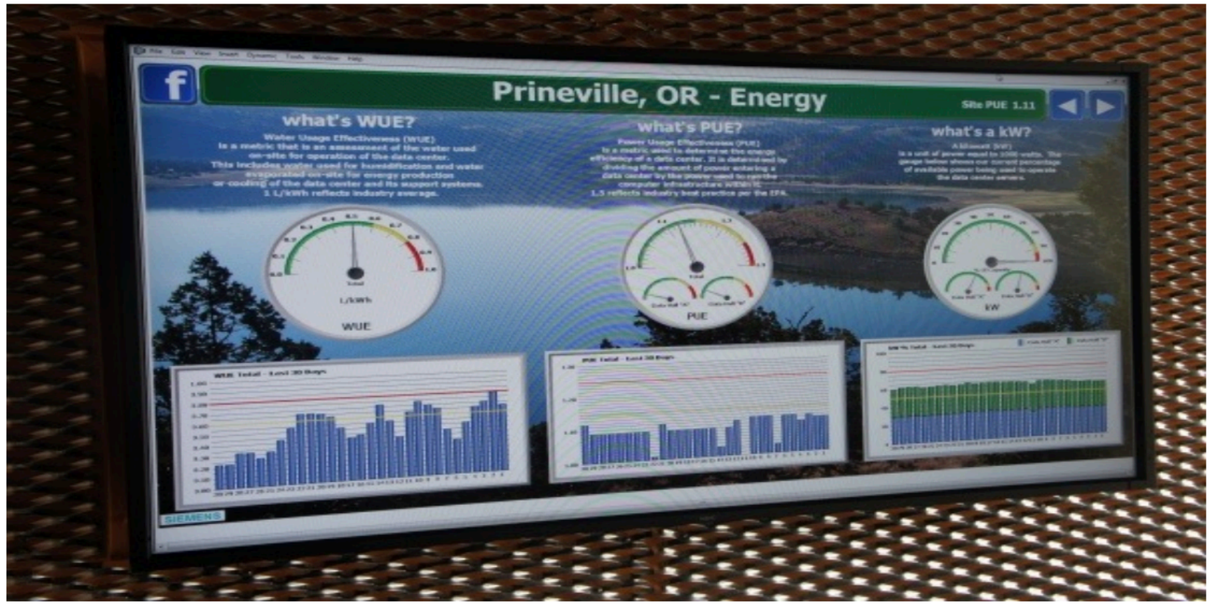




Similarities between data center locations and power plant locations?

Power

Off-site



On-site

- WUE = Annual Water Usage / IT Equipment Energy (L/kWh) – low is good
- PUE = Total facility Power / IT Equipment Power – low is good (e.g., Google~1.1)

(2) ON-DEMAND ACCESS: *AAS CLASSIFICATION

- On-demand: calling a cab vs. renting a car vs. buying one. E.g.:
  - AWS Elastic Compute Cloud (EC2): a few cents to a few $ per CPU hour
  - AWS Simple Storage Service (S3): a few cents per GB-month

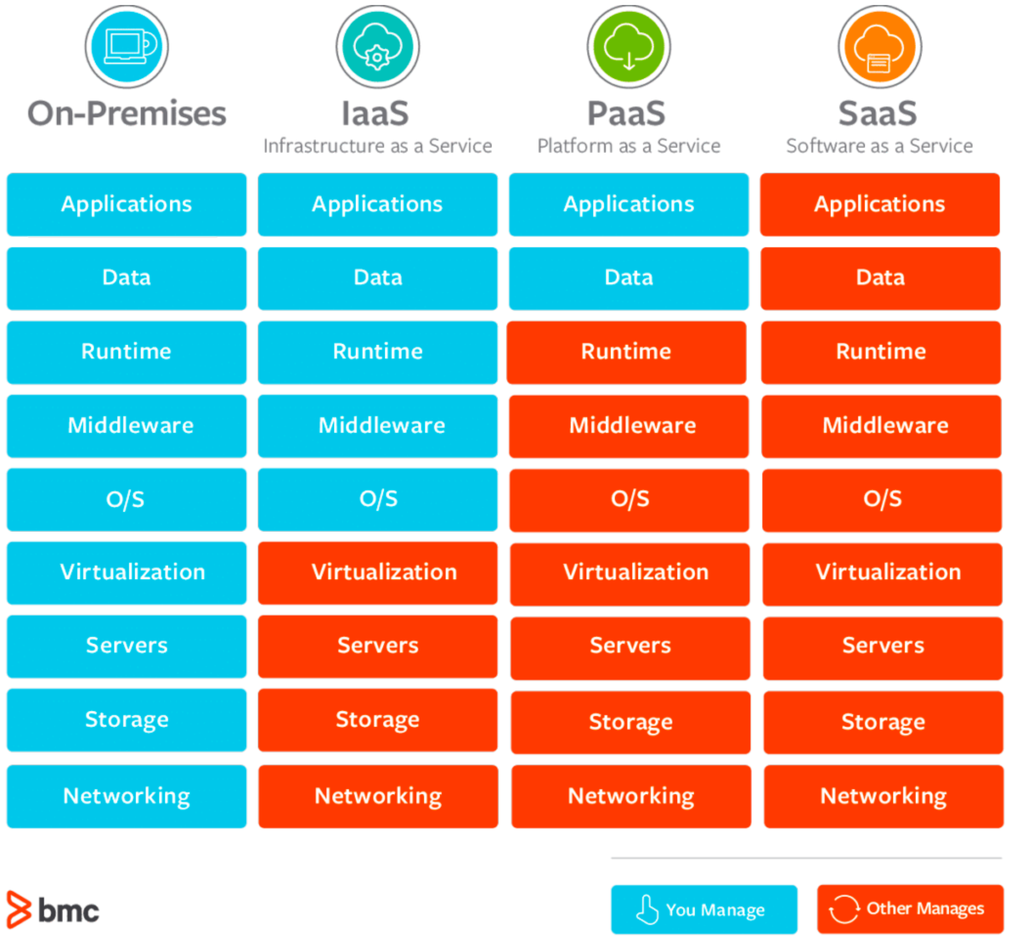
(2) ON-DEMAND ACCESS: *AAS CLASSIFICATION

- HaaS: Hardware as a Service
  - You get access to barebones hardware machines, do whatever you want with them, Ex: Your own cluster
  - Not always a good idea because of security risks
- IaaS: Infrastructure as a Service
  - You get access to flexible computing and storage infrastructure. Virtualization or containerization is one way of achieving this (cgroups, Kubernetes, Dockers, VMs,…). Often said to subsume HaaS.
  - Ex: Amazon Web Services (AWS: EC2 and S3), OpenStack, Eucalyptus, Rightscale, Microsoft Azure, Google Cloud.

(2) ON-DEMAND ACCESS: *AAS CLASSIFICATION

- PaaS: Platform as a Service
  - You get access to flexible computing and storage infrastructure, coupled with a software platform (often tightly coupled)
  - Ex: Google's AppEngine (Python, Java, Go)
- SaaS: Software as a Service
  - You get access to software services, when you need them. Often said to subsume SOA (Service Oriented Architectures).
  - Ex: Google docs, MS Office 365 Online
- And new recently: FaaS = Function as a Service
  - Ex: AWS Lambda, Azure Functions, etc.

(2) ON-DEMAND ACCESS: *AAS CLASSIFICATION



(3) DATA-INTENSIVE COMPUTING

- Computation-Intensive Computing
  - Example areas: MPI-based, High-performance computing, Grids
  - Typically run on supercomputers (e.g., NCSA Blue Waters)
- Data-Intensive
  - Typically store data at datacenters
  - Use compute nodes nearby
  - Compute nodes run computation services
- In data-intensive computing, the focus **shifts from computation to the data**: CPU utilization no longer the most important resource metric, instead I/O is (disk and/or network)

(4) NEW CLOUD PROGRAMMING PARADIGMS

- Easy to write and run applications in cloud programming paradigms:
  - Data Analytics (e.g. MapReduce): parallel programming
  - Serverless: auto-scaling, low-overhead
  - Microservice: easier for scaling and maintenance
  - ...

UNIVERSITY of VIRGINIA

TAKEAWAYS

- Clouds build on many previous generations of distributed systems
- Especially the timesharing and data processing industry of the 1960-70s.
- Need to identify unique aspects of a problem to classify it as a new cloud computing problem
- Next class: distributed system foundation, with Starbucks!

ACKNOWLEDGEMENT

THIS COURSE IS DEVELOPED HEAVILY BASED ON COURSE MATERIALS SHARED BY PROF. INDRANIL GUPTA, PROF. ROBERT MORRIS AND PROF. ROXANA GEAMBASU. MANY APPRECIATIONS FOR GENEROUSLY SHARING THEIR MATERIALS AND TEACHING INSIGHTS.