Large-scale Distributed Systems and Networks

(Storskaliga Distribuerade System och Nätverk)

Slides by Niklas Carlsson (including slides based on slides by P. Gill and Y. Shavitt)

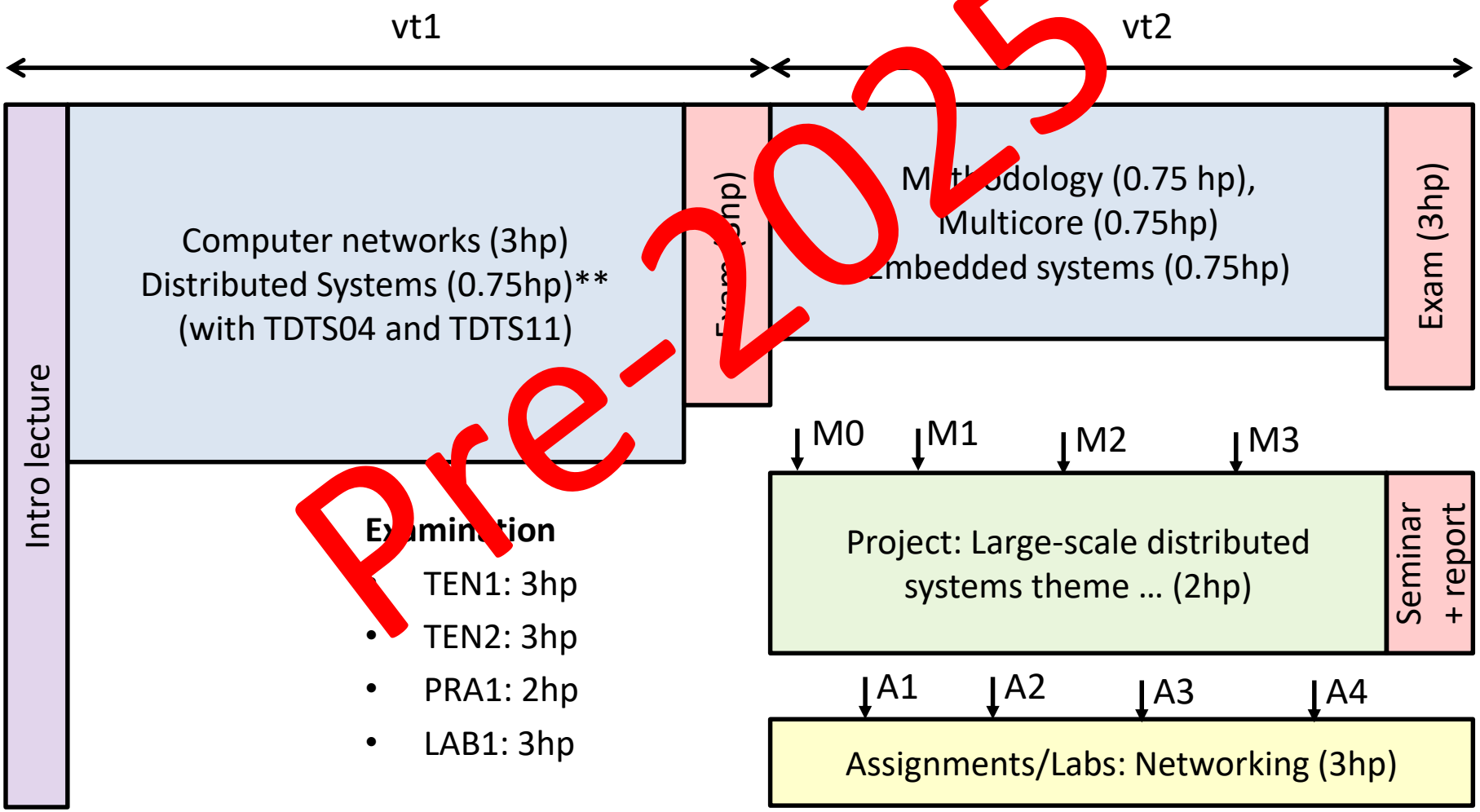
Scalability and systems thinking

- Systems thinking with focus on scalability
  - Holistic perspective (layers, components, etc.)
  - Large distributed systems and services
  - Networks and distributed systems ”hand-in-hand”
  - Single to multicore; single to million machines/users
  - Scalable methods and architectures
  - Modeling and abstraction of big systems (including some basic mathematical modeling)
- Mix of theory and practice
  - ”The knowledge is not yours until you use it”
  - Using experiments and measurements to improve the understanding of real systems in the wild + discuss the future

Subject knowledge

- Networking (vt1)
  - Basics/foundation, similar to TDTS06, TDTS11, and TDTS04 (12-14 lectures). Gives eligibility to TDTS21 (advanced networking).
  - Assignments/labs (at least one for each of the three layers 3, 4, and 5)
- Distributed systems (vt1)
  - Some introductory lectures (4 lectures)
  - Project (groups of 3-4 students)
- Multicore (vt2)
  - Kristoffer Kessler (4 lectures)
- Embedded systems (vt2)
  - Petru Eles (3 lectures)
- Methods to understand and evaluate large-scale systems (vt2)
  - Some introductory lectures (4 lectures)
  - Modeling, abstraction, and data-driven analysis methods for large-scale systems and services

Overview (2024)



Overview (2024)…

vt1

Exam	Teaching/exam block	Comment
TEN1	Networking (Ch 1-4)	Niklas (TEN1)
TEN2	Distributed systems	Niklas (TEN2) **
TEN1	Networking (Ch 5-9)	Niklas (TEN1)

vt2

Exam	Teaching/exam block	Comment
TEN2	Methodologies	Niklas (TEN2)
TEN2	Multicore	Christoff (TEN2)
TEN2	Embedded systems	Petru (TEN2)

Exam	Teaching/exam block	Comment
LAB1	Assignments	A1-A4
PRA1	Milestones	M0-M3
PRA1	Seminars	Mid+End

Pre-2025

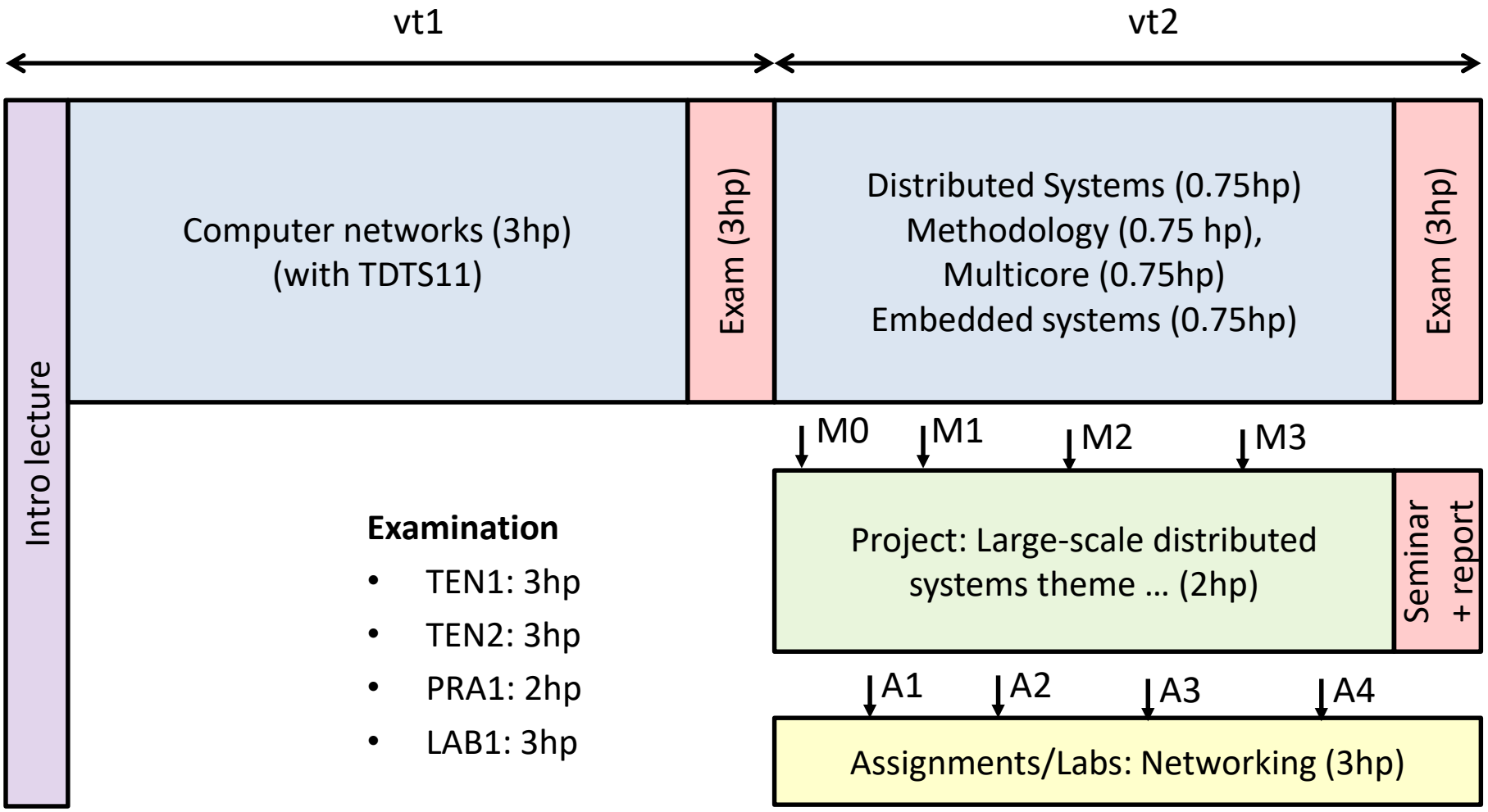
TEN1 (March 19): 3 ECTS

TEN2 (May 31): 3 ECTS

LAB1 (vt2): 3 ECTS

PRA1 (vt2): 2 ECTS

Overview (2025)



Examination

- TEN1: 3hp
- TEN2: 3hp
- PRA1: 2hp
- LAB1: 3hp

Overview (2025)…

vt1

Exam	Teaching/exam block	Comment
TEN1	Networking (Ch 1-9)	Niklas (TEN1)

TEN1 (March 24): 3 ECTS

vt2

Exam	Teaching/exam block	Comment
TEN2	Distributed systems	Niklas (TEN2)
TEN2	Methodologies	Niklas (TEN2)
TEN2	Multicore	Christoff (TEN2)
TEN2	Embedded systems	Petru (TEN2)

TEN2 (June 3): 3 ECTS

Exam	Teaching/exam block	Comment
LAB1	Assignments	A1-A4
PRA1	Milesones	M0-M3
PRA1	Seminars	Mid+End

LAB1 (vt2): 3 ECTS

PRA1 (vt2): 2 ECTS

Projects and assignments

- **2025: Practice working in teams of 2-4 students.**

- Assignments (and lab sessions)
  - Groups of 2 students
  - Register in webreg by Friday (Apr. 34, 2025)
  - 4 assignments:
    - Split across multiple network layers
    - 2 x wireshark (HTTP + TCP), proxy, and DV
  - Note: Many advantages having labs during vt2 (instead of vt1) – actually think it is better! Try to use them to finish the labs quicker.
- Project
  - Groups of 3-4 students (larger groups than past courses you have seen)
  - Clear ”milestones” introducing both incremental and iterative report writing, as well as oral presentation
  - Multiple ”milestones” with ”peer reviewing”
  - Register for webreg by Friday (Apr. 4, 2025)
  - Projects released by Monday (Apr. 7, 2025)
  - Request projects by Thursday (priority if on Wednesday)

Course evaluations [focus on complaints]

- Different quality of lectures: For those with most complaints, there are online alternatives + excellent textbook. Also, added new instructor and changed distribution of lectures.
- Shared summary lecture confusing [2023]: Separated + Niklas gives lecture for TDDE35
- vt1 + vt2 split [*Per design. See prior slides + explanations/motivations*]
  - Student prioritize Pintos + envar. [*Explain/motivate*]
  - [some] Time consuming [*Explain/motivate*]
  - Some suggestions [mostly not feasible/reasonable]
  - Did move DS lectures to vt2 (2025); further planned
- Expected attendance at seminars (2+4+2+4 hrs): *Not much time + Important learning opportunity*
- Request for help/suggestions how to succeed in TDDE35 [*Added more examples + pointers*]
- Some comments may have been a personal (defense) response

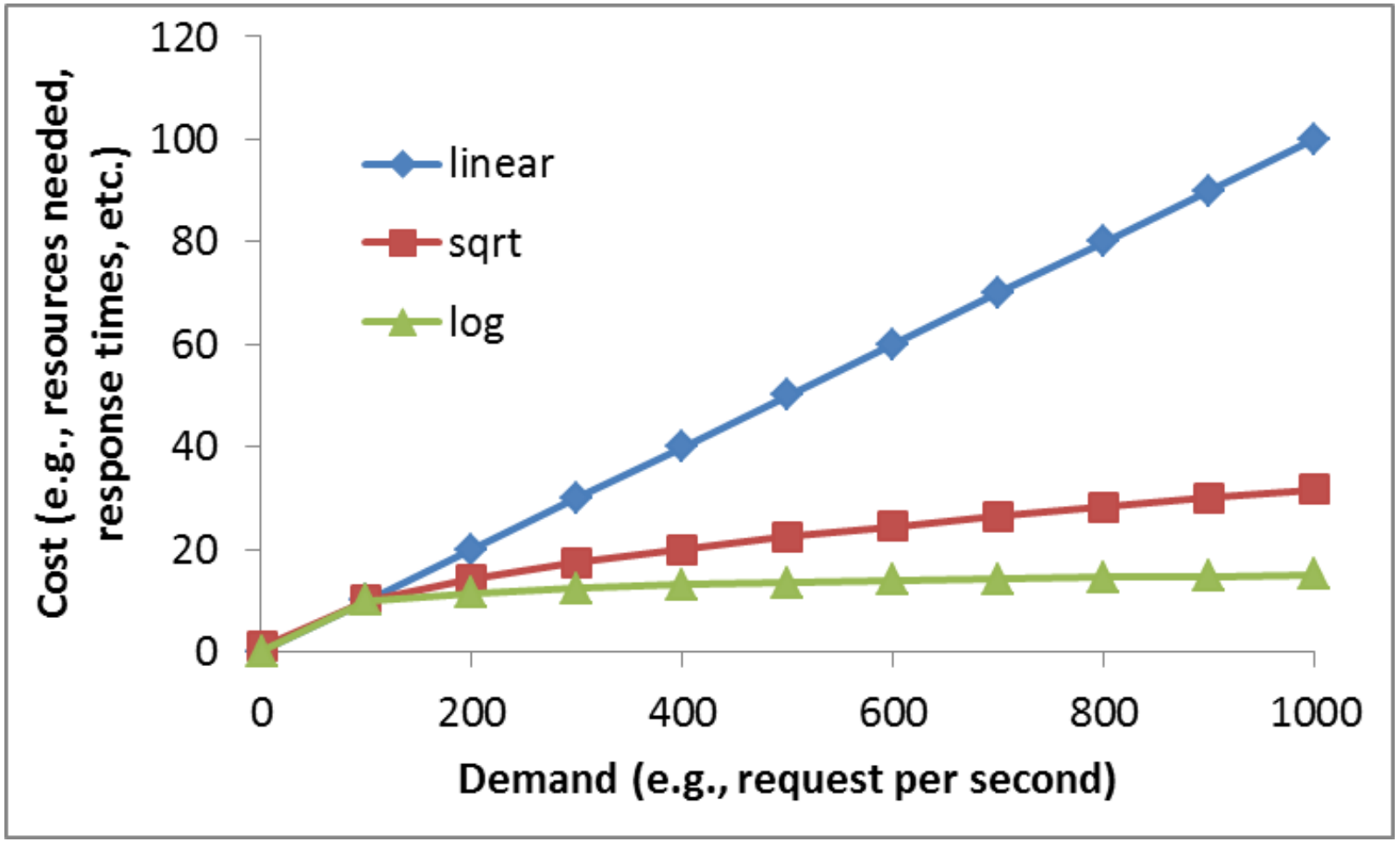
Reminder: Attend lectures …

- In class: Examples to help build an understanding and intuition for ``scalability" and ``system thinking"
  - These abilities are hard-to-impossible to learn only from notes!!
- Projects and expectations around the projects (and report/article writing in general) will be discussed in class
- Please attend the lectures (and obtain such information …)

Scalability

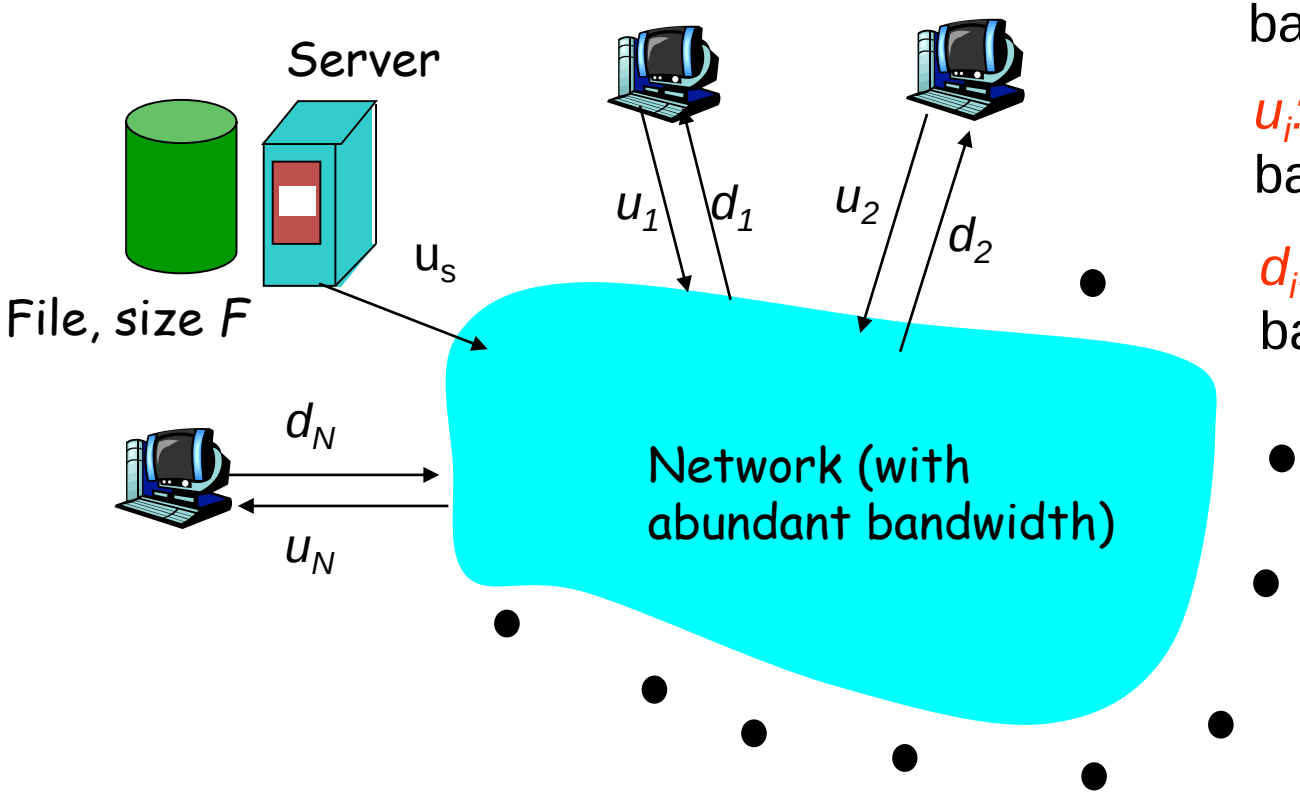
- Typically want solutions that ``scales”
  - Ability of a system, network, or process to handle a growing amount of work effectively
  - Capability to increase its total output under an increased load when resources are added
- Typically want:
  - the costs or resource capacity needed to scale sub-linearly with demand, or
  - the performance to improve at least proportionally to the capacity added

Scalability examples



Examples from earlier in the course …

Question : How much time to distribute file from one server to *N peers*?

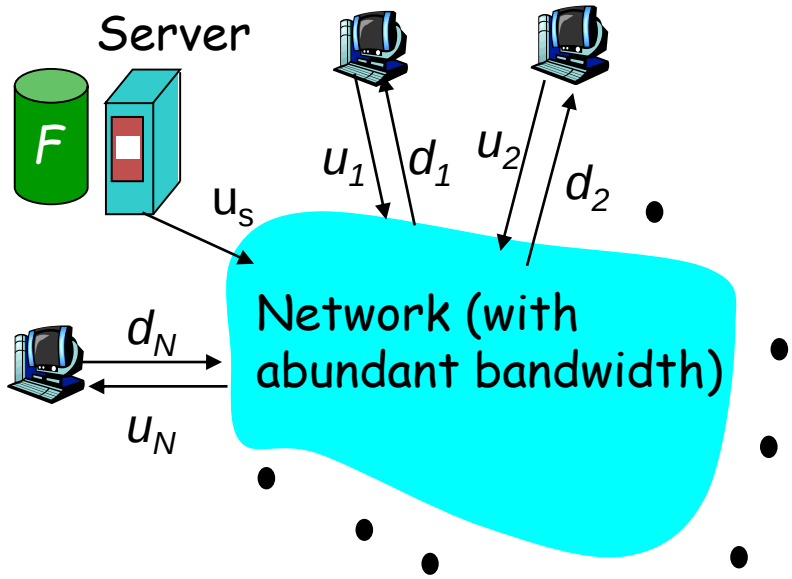


u_s: server upload bandwidth

u_i: peer i upload bandwidth

d_i: peer i download bandwidth

File distribution time: server-client



Time to distribute F to N clients using client/server approach

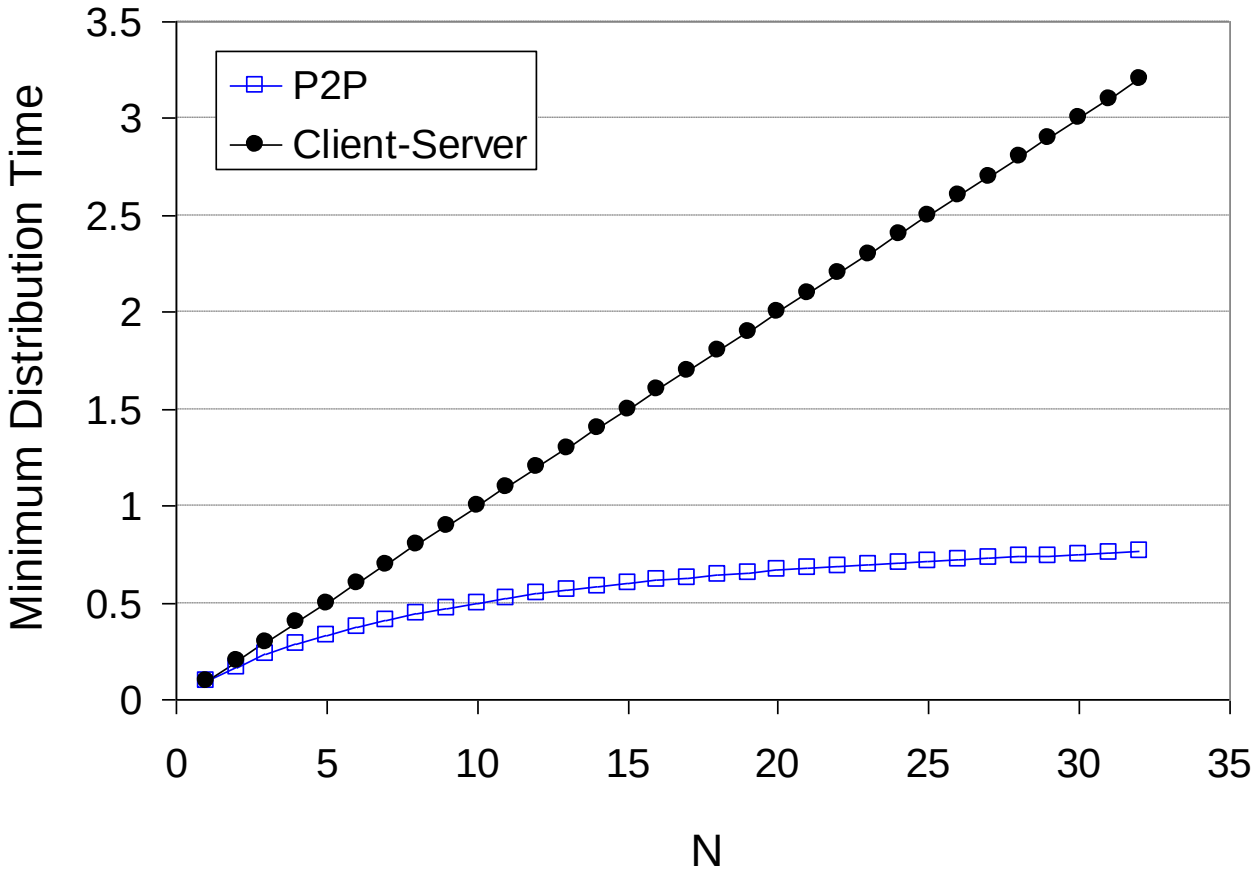
$$= d_{cs} = \max \{ NF/u_s, F/\min(d_i) \}$$

… and using a P2P approach

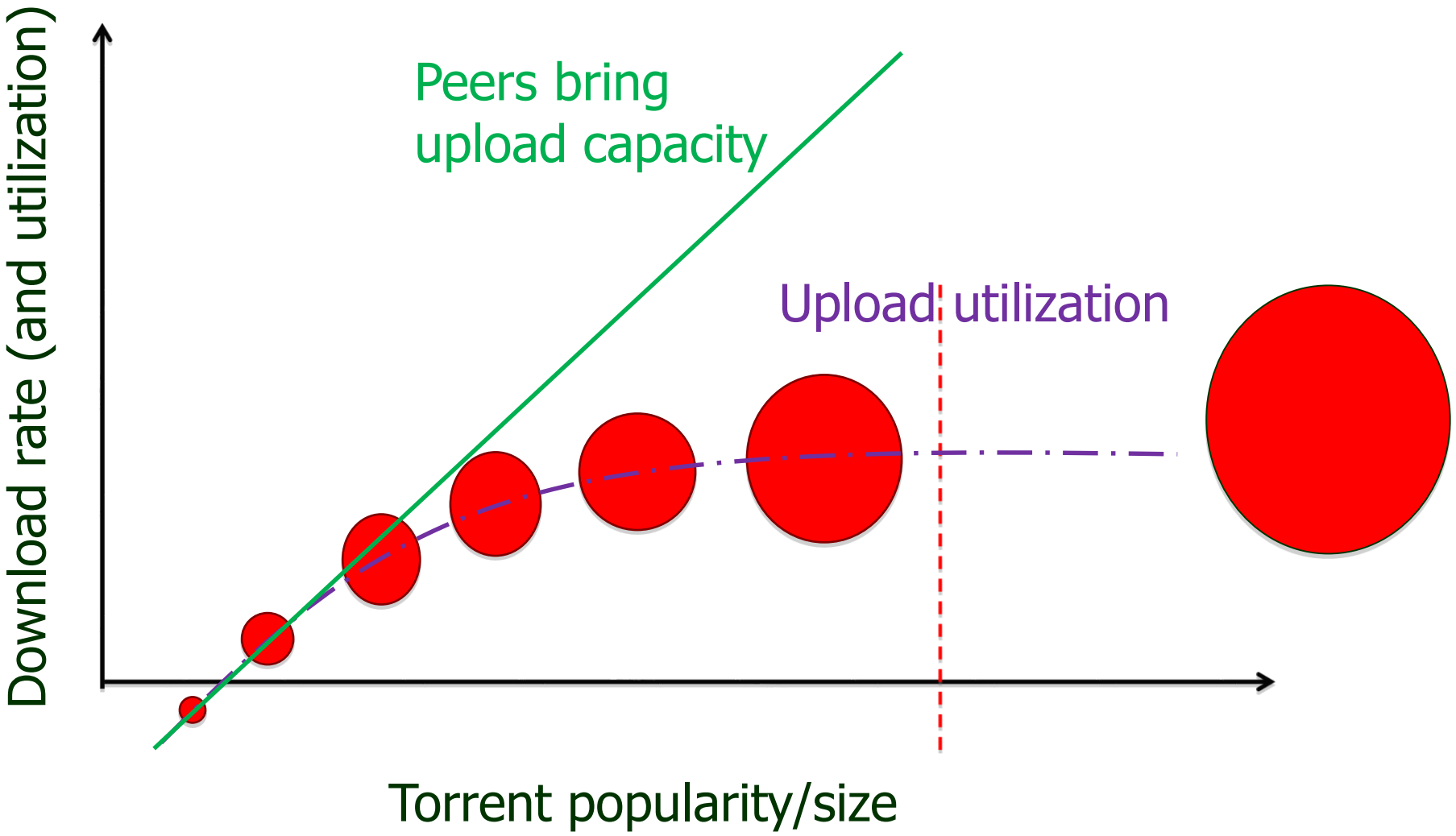
$$d_{P2P} = \max \{ F/u_s, F/\min(d_i) , NF/(u_s + \Sigma u_i) \}$$

Server-client vs. P2P: example

Client upload rate = u, F/u = 1 hour, u_s = 10u, $d_{min} \geq u_s$



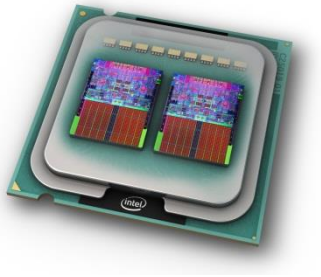
Similarly, BitTorrent upload utilization …



… more examples later …

Systems thinking

- We want to understand the full system and the ecosystem it operates within; e.g.,
  - Understanding the full system
  - Looking at the parts and how they interact
- This course provide many examples ...



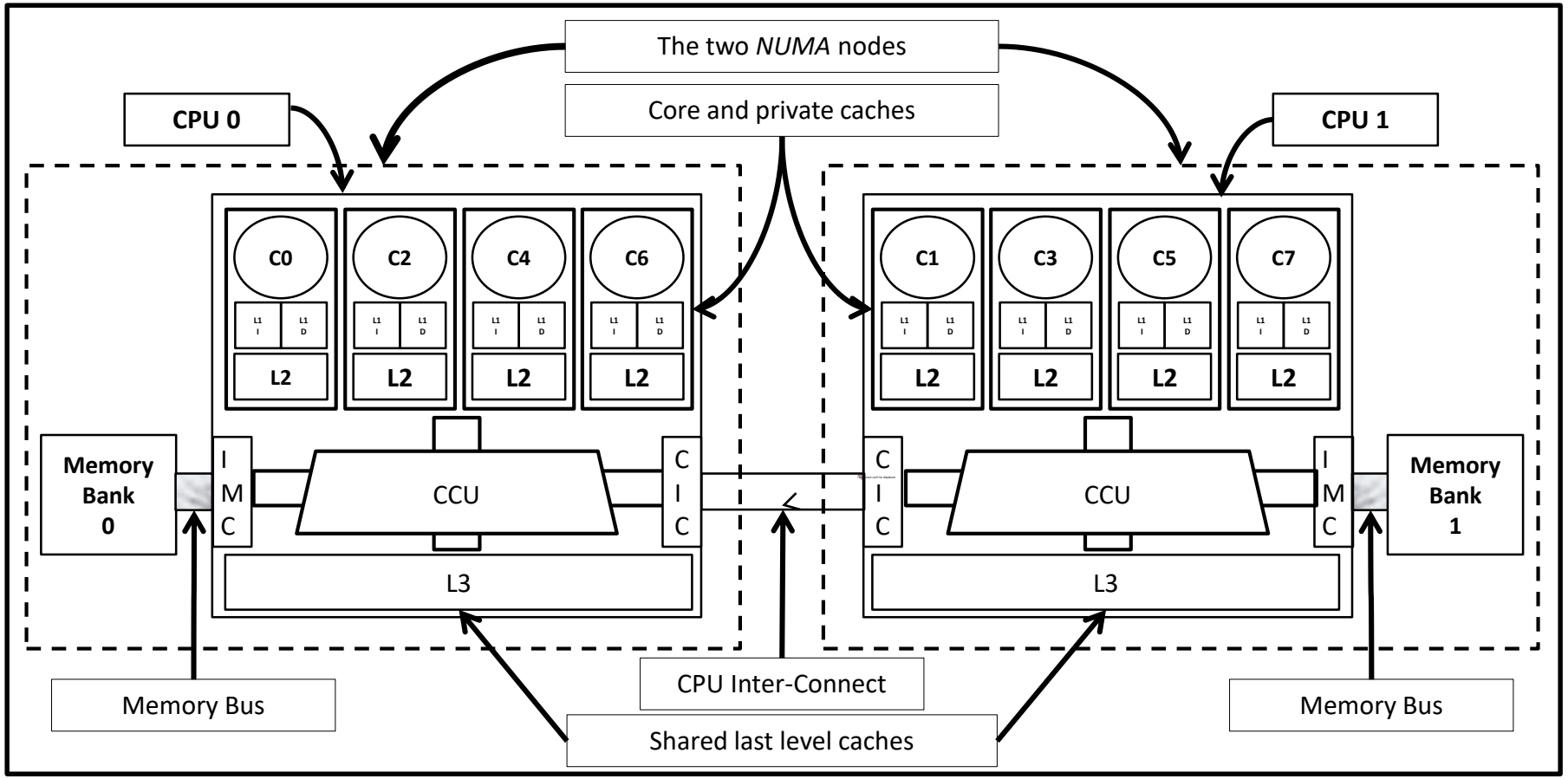
Measurements

- It has often been stated that
  - "you can't manage what you can't measure" ...
- Effective tool to understand, model, test, and improve existing systems …
  - E.g., often want to identify (and fix) system bottlenecks

Multicore systems

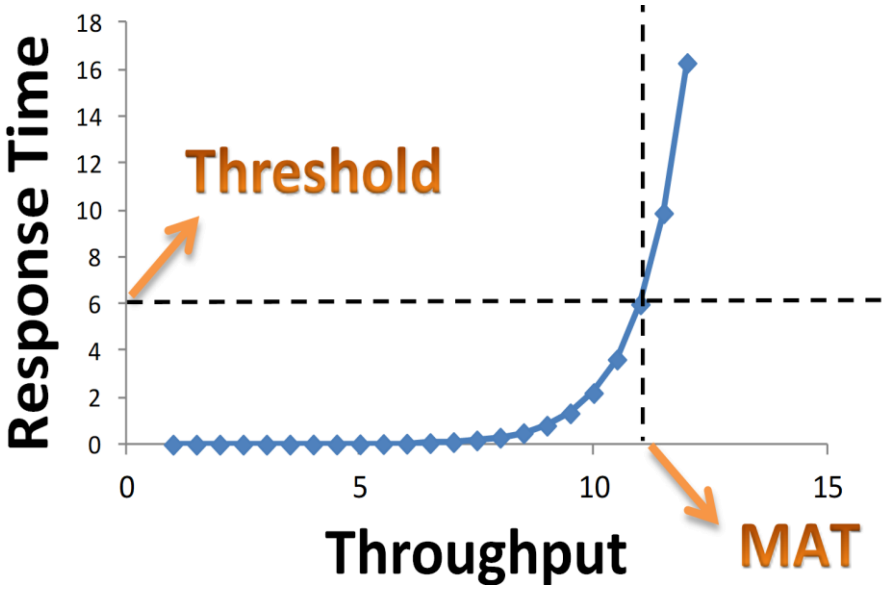
NUMA Architecture

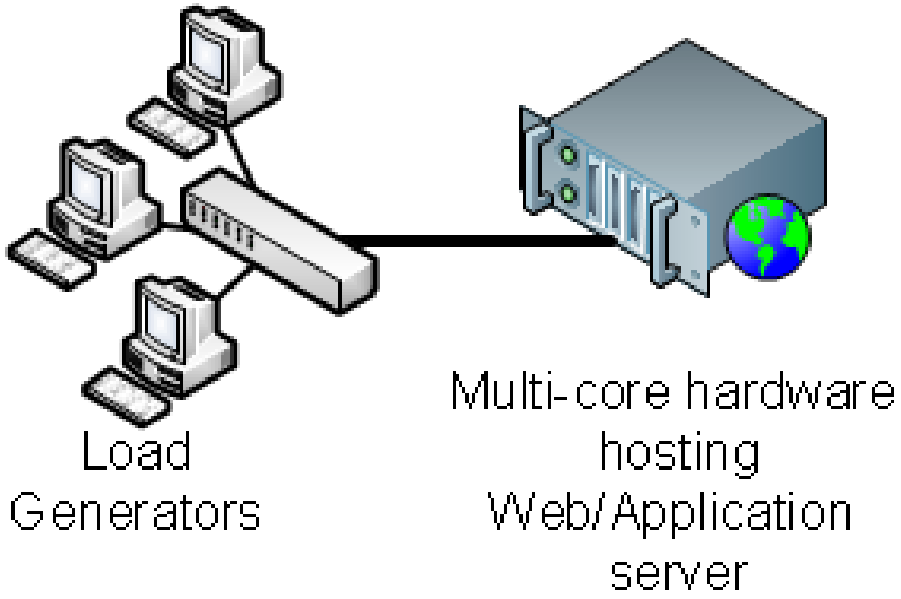
An example of a two processor eight core NUMA system



Scalability Evaluation Measurements

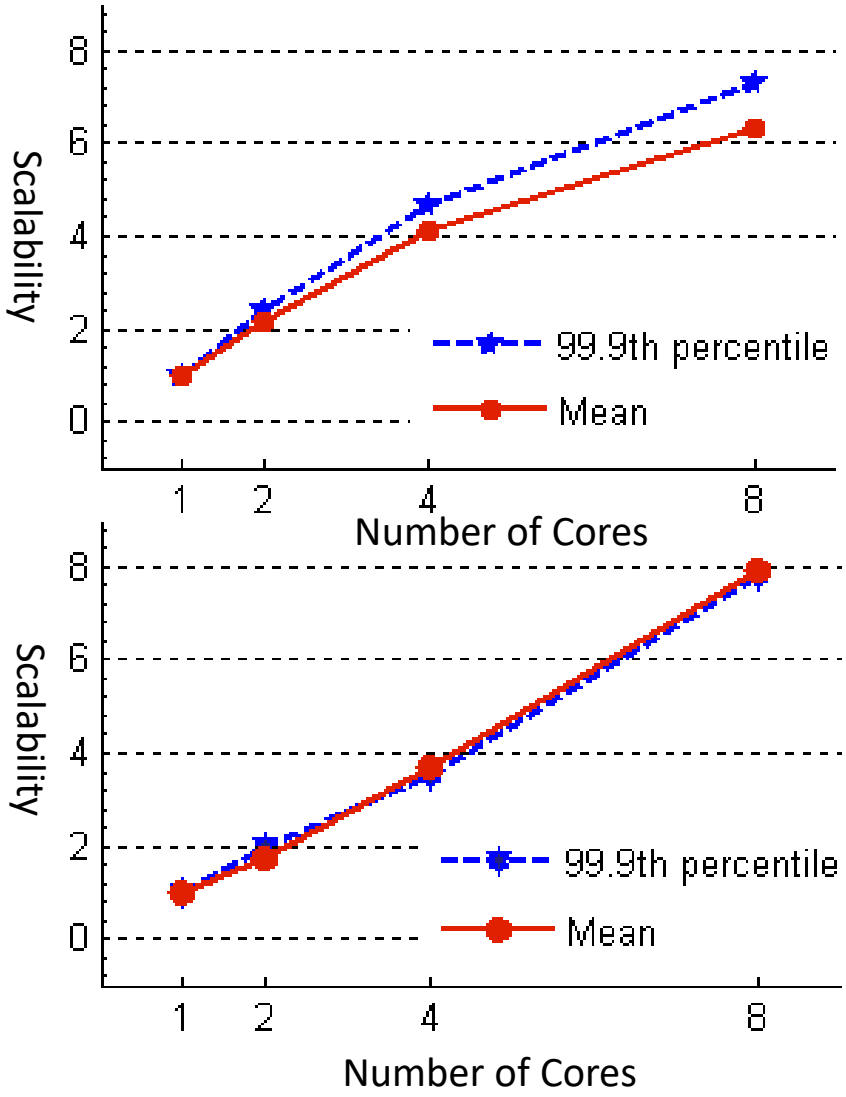
- E.g., Measure Web server scalability for workloads [ICPE '13]
  - Typically want to provide some 99% response time
  - Example scalability measure: Maximum Achievable Throughput (MAT)





RESULTS

- TCP/IP Intensive workload
  - Sub-linear
    - Maximum Achievable Throughput
      - 146,000 req/sec
- SPECweb Support workload
  - Almost linear
    - Maximum Achievable Throughput
      - 23,000 req/sec

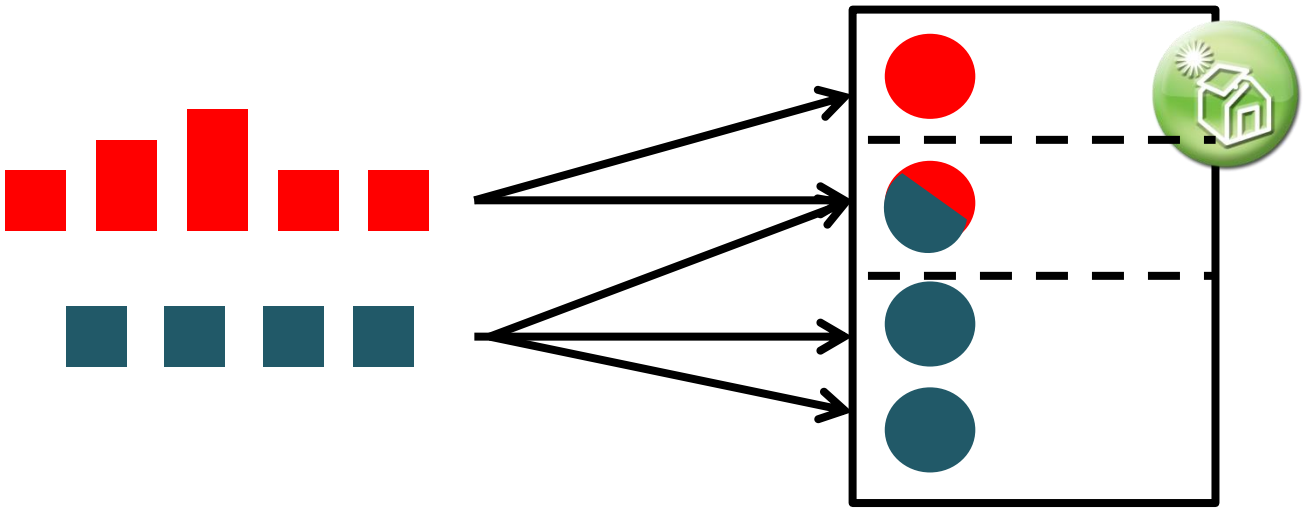


Identification of bottlenecks

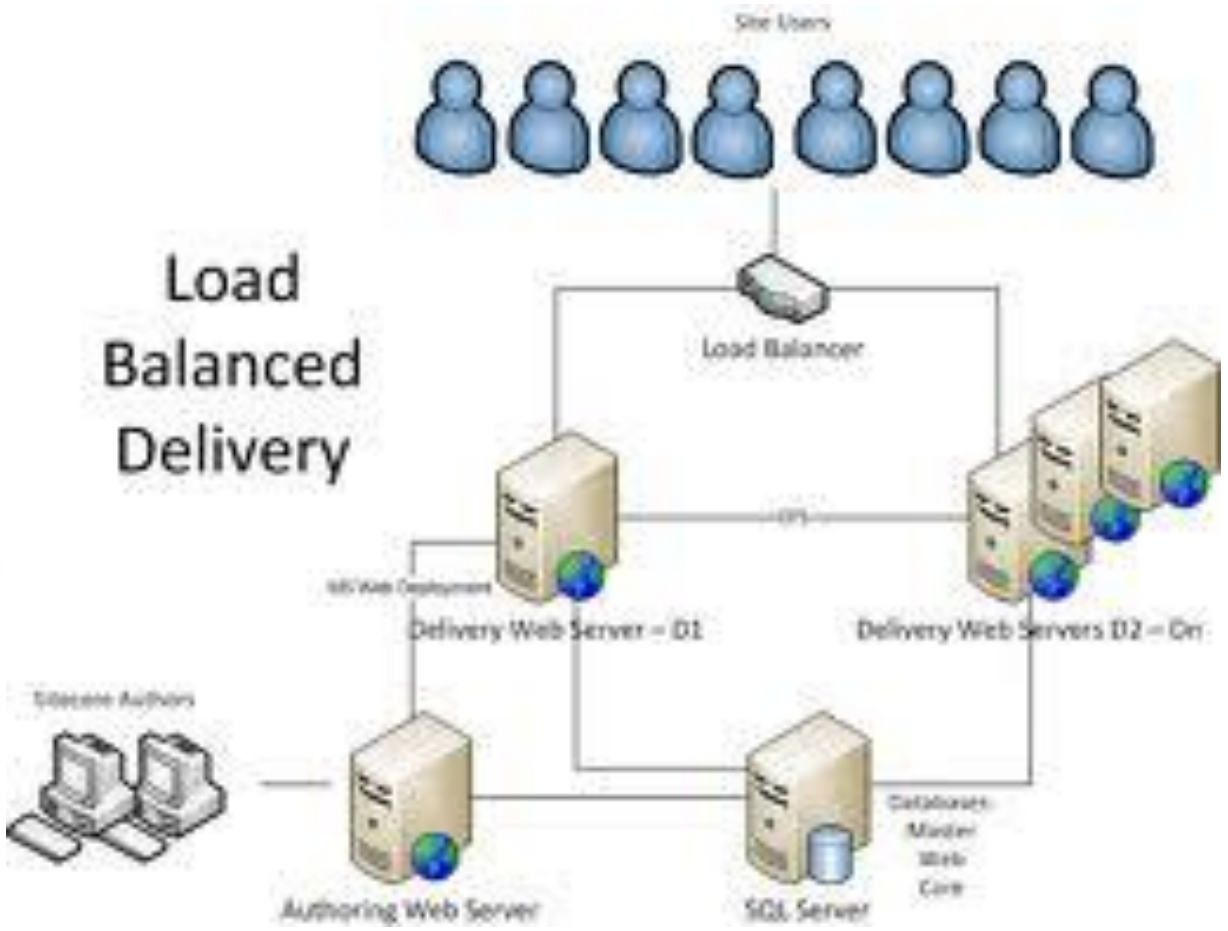
- E.g., memory, CPU, network, cache hierarchy, interconnect bus, scheduler, …
  - Black-box testing
  - Low-level instrumentation

Identification of bottlenecks

- E.g., memory, CPU, network, cache hierarchy, interconnect bus, scheduler, …
  - Black-box testing
  - Low-level instrumentation
- Multiple workloads …



Often many servers (and racks)



… and data centers …

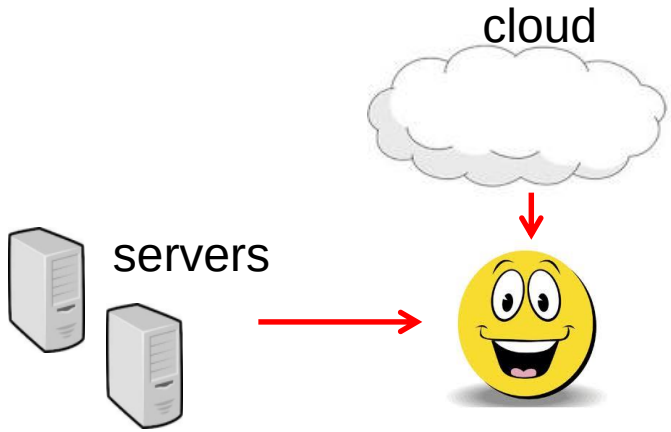
… cost-efficient delivery …

… and different flexibility …

- Minimize content delivery costs

	Bandwidth	Cost
Cloud-based	Elastic/flexible	$$$
Dedicated servers	Capped	$

How to get the best of two worlds?



… and from who?