中山大學 SUN YAT-SEN UNIVERSITY

国家超级计算广州中心 NATIONAL SUPERCOMPUTER CENTER IN GUANGZHOU

Computer Architecture

计 算 机 体 系 结 构

第21讲：WSC & Interconnect （2）

张献伟

xianweiz.github.io

DCS3013, 12/14/2022

Quiz Questions

Please email to zhangxw79@mail.sysu.edu.cn (ddl: 14:40).

- Q1: for MSI snooping, how to change state from S to M?

  Invalidate all other copies --exclusive--> update value in cache

- Q2: for directory-based protocol, how to reduce communication overhead?

  Intervention forwarding, request forwarding, parallelization …

- Q3: differences between coherence and consistency?

  Same vs different location, eventually vs when, cache vs. mem, …

- Q4: what are possible values of *data* in TSO processors? Give the ordering.

  1: ①②③④

P0	P1
// flag = 0; data = 0;	
data = 1; ① flag = 1; ②	while (flag == 0); ③ print data; ④

- Q5: what about PSO processors?

  1: ①②③④/②①③④/②③①④

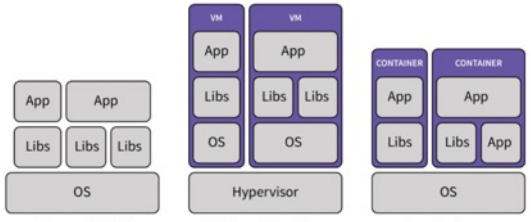
  0: ②③④①

Warehouse-scale Computer[仓储规模]

- Massive scale datacenters: 10,000 to 100,000 servers + networks to connect them together
  - Emphasize cost-efficiency
  - Attention to power: distribution and cooling
  - (relatively) homogeneous hardware/software
- Single gigantic machine
- Offer very large applications (Internet services): search, voice search (Siri), social networks, video sharing
- Very highly available: < 1 hour down/year
  - Must cope with failures common at scale
- "…WSCs are no less worthy of the expertise of computer systems architects than any other class of machines" (Barroso and Hoelzle, 2009)

Warehouse-scale Computer (cont.)

- Differences with **HPC** “clusters”[高性能集群]:
  - Clusters have higher performance processors and network
    - HPC apps are more interdependent and communicate more frequently
  - Clusters emphasize TLP and DLP, WSCs emphasize RLP
    - HPC emphasizes latency to complete a single task vs. bandwidth to complete many independent tasks
    - HPC clusters tend to have long-run jobs that keep servers fully utilized
- Differences with **datacenters**[数据中心]:
  - Datacenters consolidate different machines and software into one location
  - Datacenters emphasize virtual machines and hardware heterogeneity in order to serve varied customers



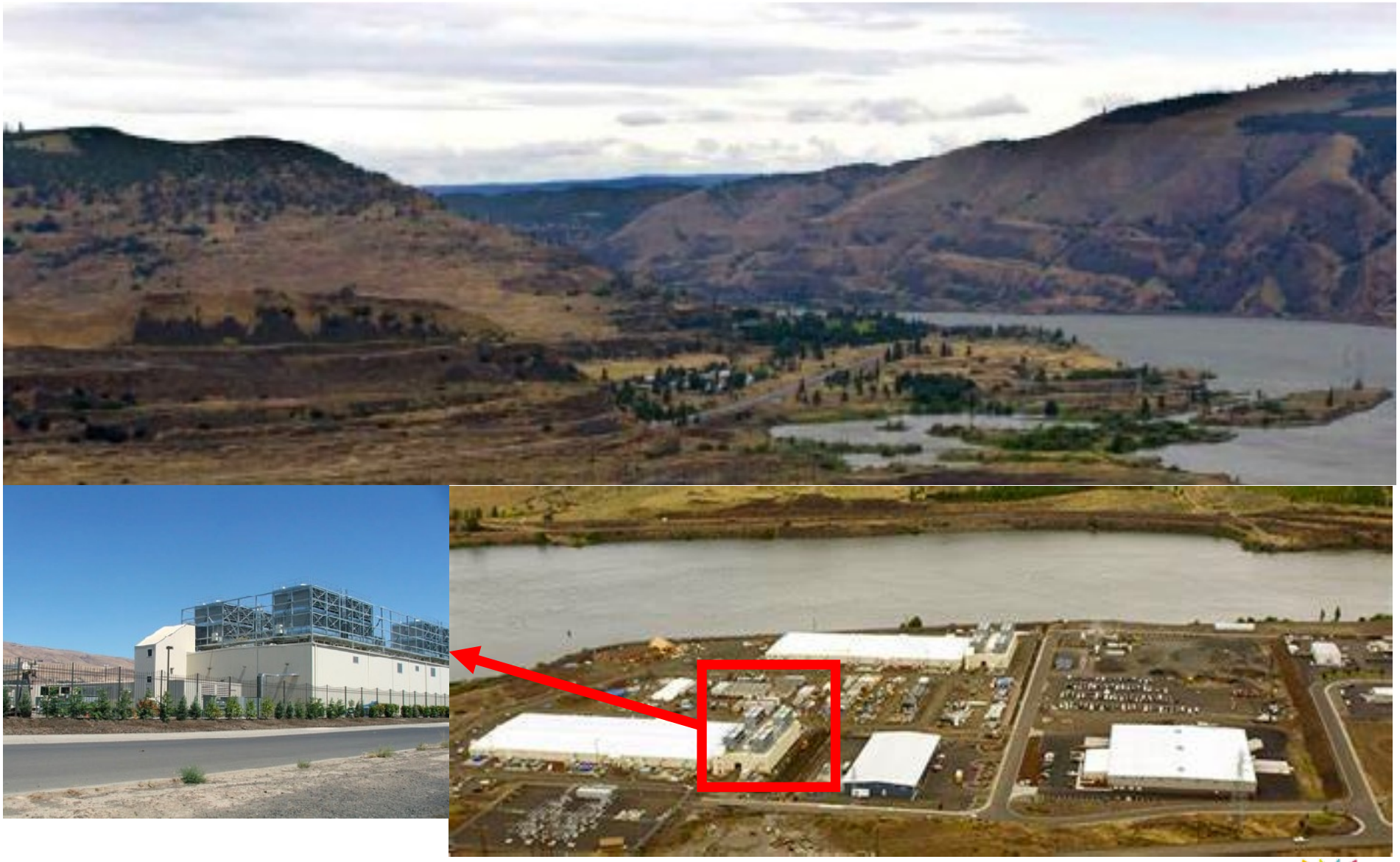
Design Goals of WSC[设计目标]

- WSCs share many goals and requirements with servers
  - Cost-performance
    - Work done per $
  - Energy efficiency
    - Work done per J
  - Dependability via redundancy
    - 99.99% of availability, i.e., less 1h down per year
  - Network I/O
    - Good interface to external world
  - Both interactive and batch processing workloads
    - Interactive: e.g., search and social networking with Billions of users
    - Batch: calculate metadata useful to such services, e.g., MapReduce jobs to convert crawled pages into search indices

Design Goals of WSC (cont.)

- Unique to WSCs
  - Ample parallelism
    - Batch apps: many independent data sets with independent processing (Data-Level and Request-Level Parallelism)
  - Scale and its opportunities/problems
    - Relatively small number of WSC make design cost expensive and difficult to amortize
    - But price breaks are possible from purchases of very large numbers of commodity servers
    - Must also prepare for high component failures
  - Operational costs count
    - Cost of equipment purchases << cost of ownership
  - Location counts
  - Computing efficiently at low utilization
    - WSC servers are rarely fully utilized

Google's Oregon WSC



https://inst.eecs.berkeley.edu/~cs61c/resources/su18_lec/Lecture21.pdf

Containers in WSCs[集装箱]

Inside WSC

Inside Container

Programming Models for WSCs[编程模型]

- Batch processing framework: MapReduce
  - The MapReduce runtime environment schedules map tasks and reduce tasks to the nodes of a WSC
  - MapReduce can be thought of as a generalization of the SIMD operation
    - Except that a function to be applied is passed to the data
- Map: (in_key, in_value) → list(interm_key, interm_val)
  - Slice data into "shards" or "splits" and distribute to workers
  - Compute set of intermediate key/value pairs
- Reduce: (interm_key, list(interm_value)) → list(out_value)
  - Combines all intermediate values for a particular key
  - Produces a set of merged output values (usually just one)

MapReduce Example

- Map phase: (doc name, doc contents) → list(word, count)

// “I do I learn”” → [(“I”,1),(“do”,1),(“I”,1),(“learn”,1)]

```
map(key, value):
    for each word w in value:
      emit(w, 1)
```

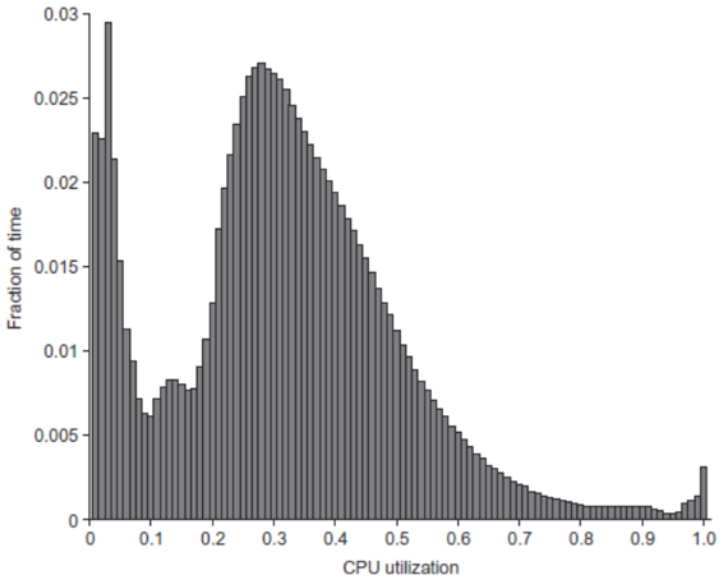
- Reduce phase: (word, list(count)) → (word, count_sum)

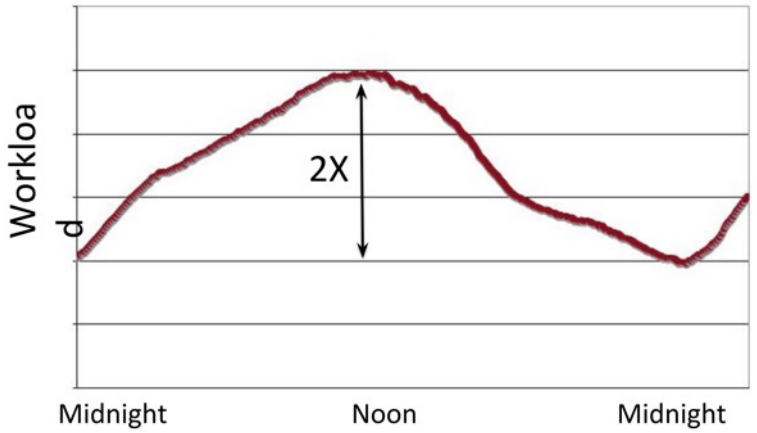
// (“I”, [1,1]) → (“I”,2)

```
reduce(key, values):
    result = 0
    for each v in values:
       result += v
    emit(key, result)
```

WSC Software[软件]

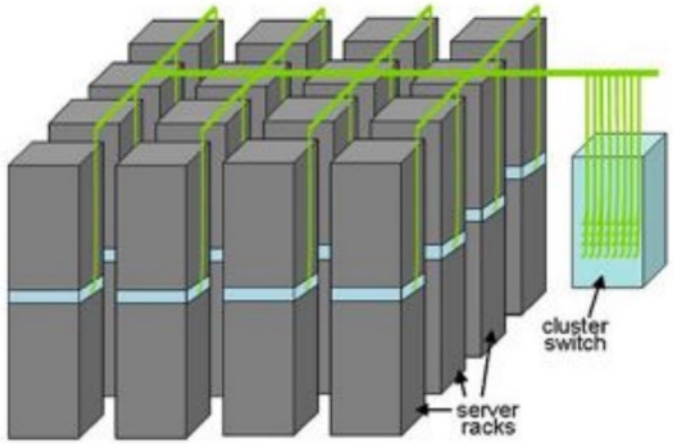
- Must scale up and down gracefully in response to varying demands
  - Varying workloads impact availability
- Must cope with failures gracefully
  - High failure rate impact reliability and availability
- More elaborate hierarchy of memories, failure tolerance, workload accommodation makes WSC software development more challenging than software for single computer



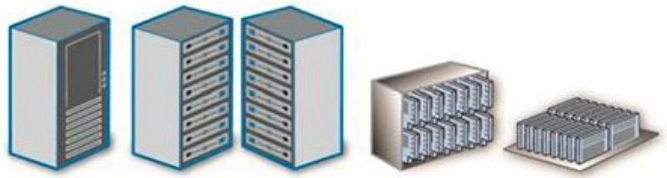


Equipment Inside a WSC

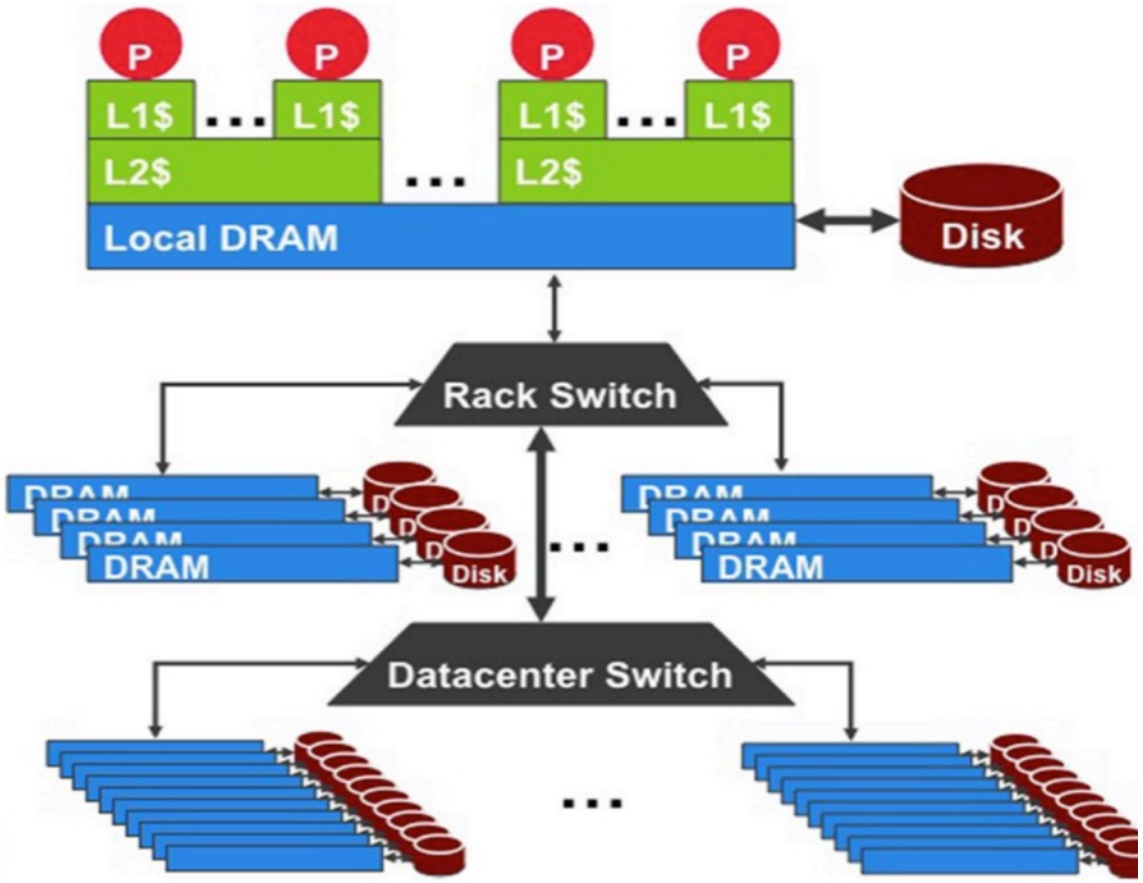
- **Server**[服务器]
  - 1 ¾ inches high “1U”（4.445cm）
  - 8 cores, 16 GB DRAM, 4x1 TB disk
- **Rack**[机架]
  - 7 feet（213.36cm）
  - 40-80 servers + Ethernet local area network (1-10 Gbps) switch in middle (“rack switch”)
- **Array** (a.k.a., cluster)[集群]
  - 16-32 server racks + larger local area network switch (“array switch”)
    - Expensive switch (10X bandwidth, 100x cost)



Server, Rack, Array



WSC Architecture[架构]

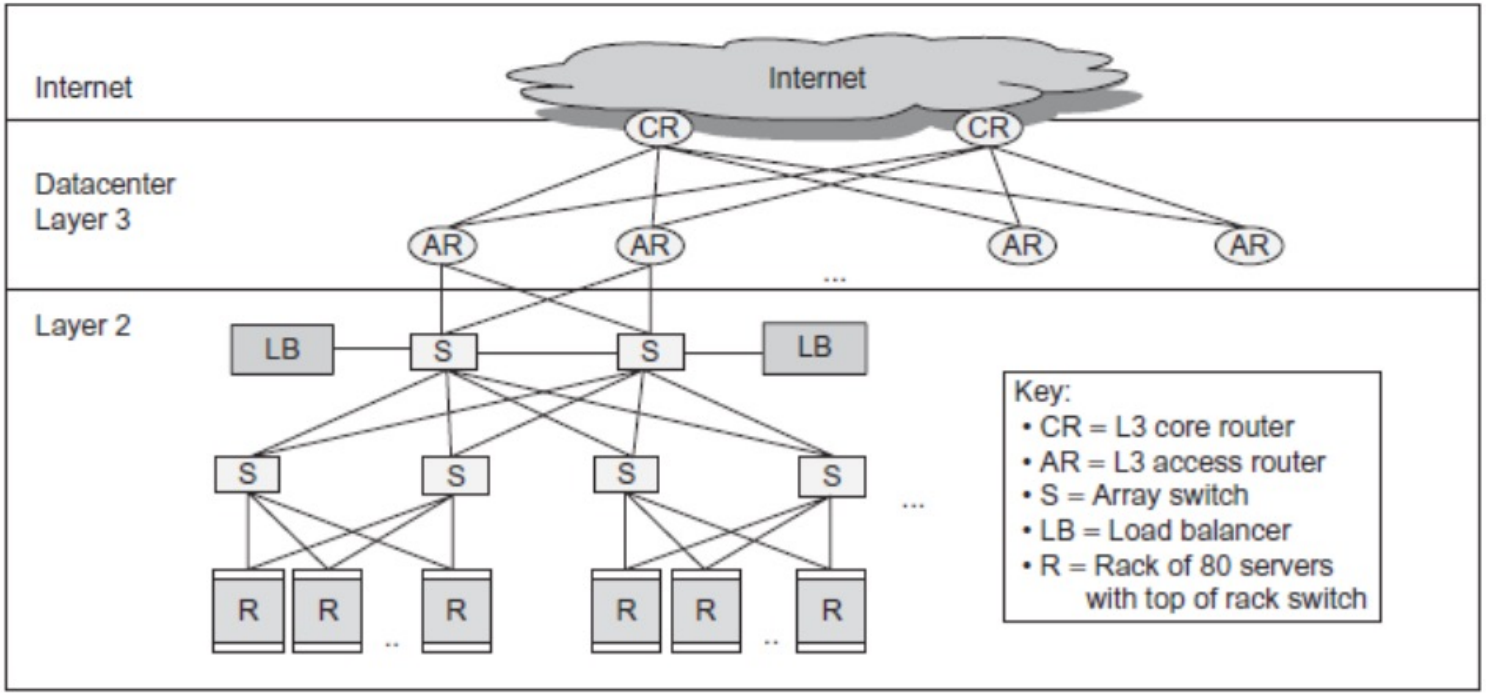


- 1U Server:
  - DRAM: 64GB, 100ns
  - Disk: 10TB, 10ms
- Rack (80 severs):
  - DRAM: 5TB, 300µs
  - Disk: 800TB, 11ms
- Array (30 racks):
  - DRAM: 150TB, 500µs
  - Disk: 24PB, 12ms

Lower latency to DRAM in another server than local disk
Higher bandwidth to local disk than to DRAM in another server

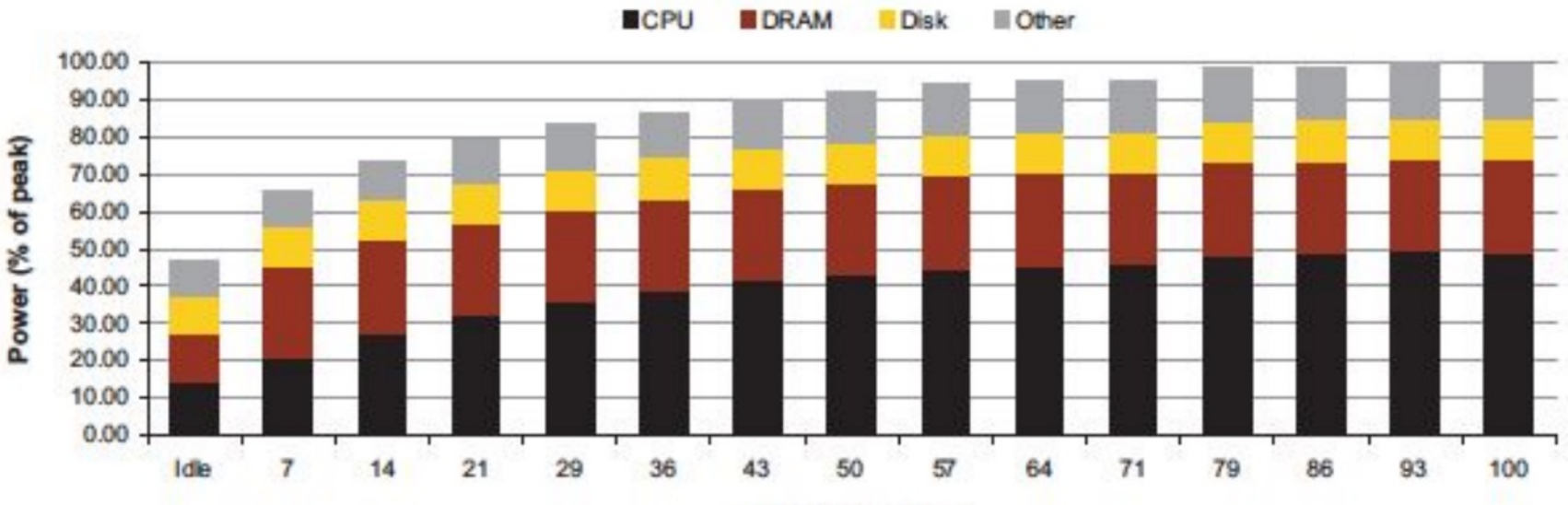
Network[网络]

- The WSC needs 40 arrays to reach 100K servers
  - One more level in the networking hierarchy
- Conventionally, Layer 3 routers to connect the arrays together and to the Internet



Power vs. Server Utilization[能耗]

- Figure: server power usage as load varies idle to 100%
- Uses ½ peak power when idle!
- Uses ⅔ peak power when 10% utilized! 90%@ 50%!
- Most servers in WSC utilized 10% to 50%
- Goal should be Energy-Proportionality: % peak load = % peak energy



https://inst.eecs.berkeley.edu/~cs61c/resources/su18_lec/Lecture21.pdf

Power Usage Effectiveness[电源使用效率]

- Overall WSC Energy Efficiency: amount of computational work performed divided by the total energy used in the process

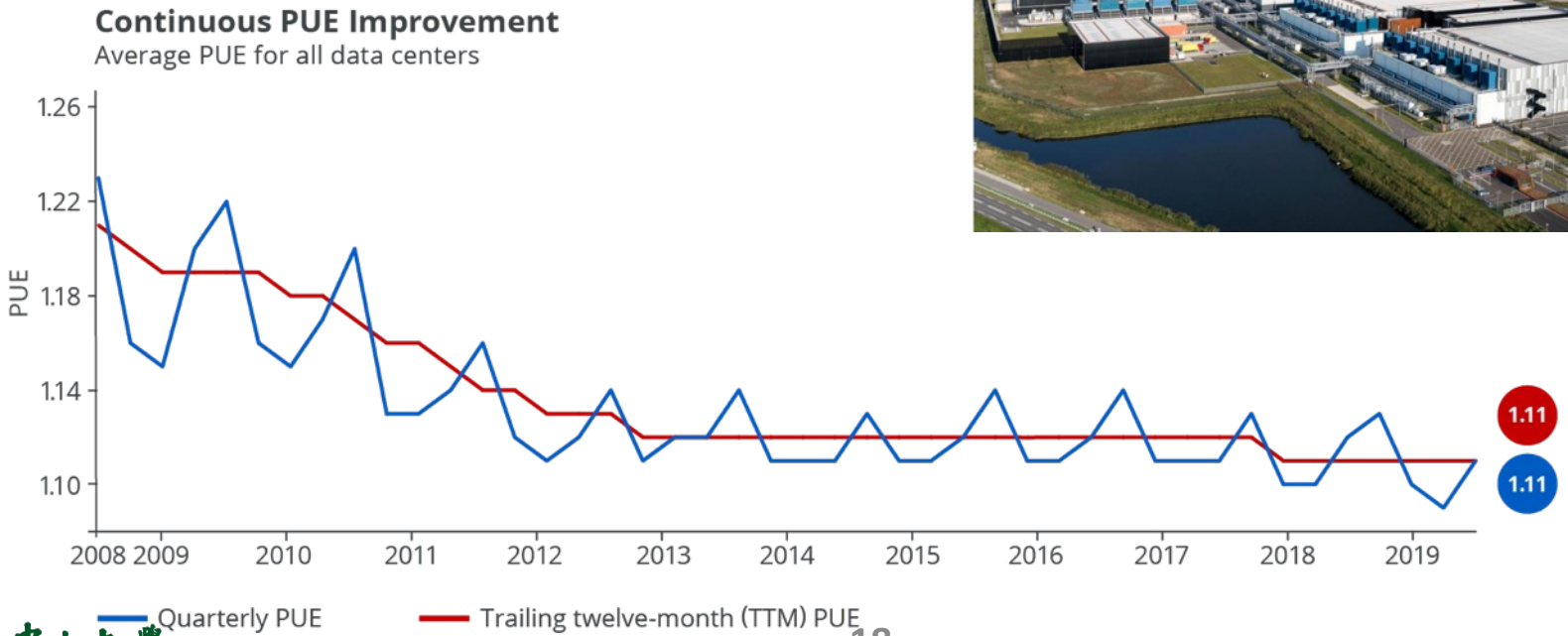
- Power Usage Effectiveness (PUE):

$$\frac{\text{Total Building Power}}{\text{IT equipment Power}}$$

  - Power efficiency measure for WSC, not including efficiency of servers, networking gear
  - Power usage for non-IT equipment increases PUE
  - 1.0 is perfection, higher numbers are worse
  - Google WSC's PUE: 1.2

Power Usage Effectiveness (cont.)

- Average PUE of the 15 google WSCs 2008 – 2017
- Google's Belgium WSC PUE: 1.09
  - Careful air flow handling
  - Elevated cold aisle temperatures
  - Use of free cooling
  - Per-server 12-V DC UPS

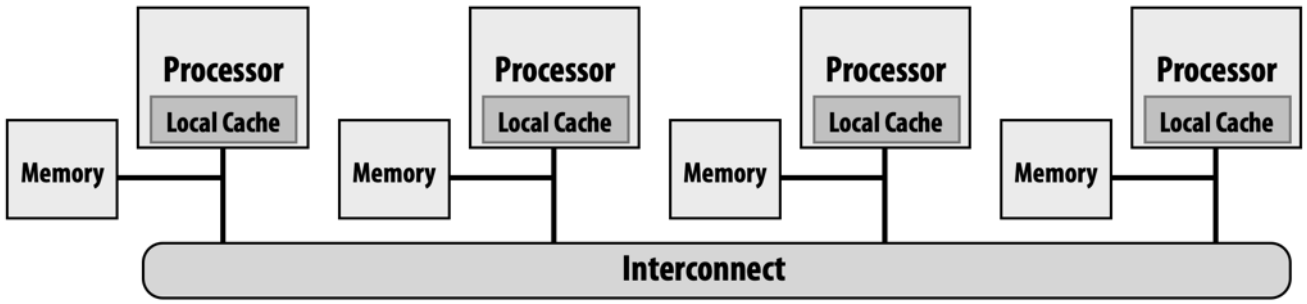


https://inst.eecs.berkeley.edu/~cs61c/resources/su18_lec/Lecture21.pdf

Interconnection Network

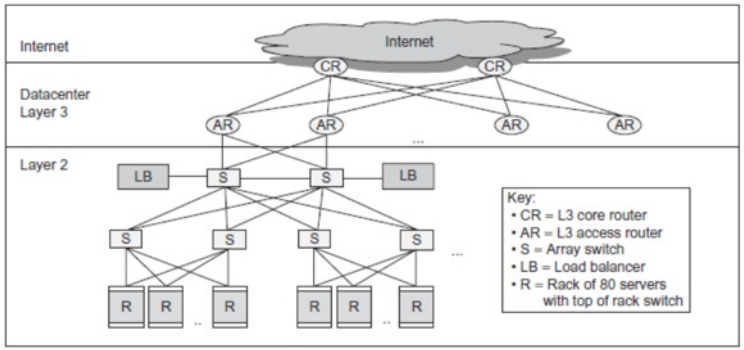
Interconnection Networks[互联网络]

- An **Interconnection Network** (ICN) is a programmable system that transports data between terminals
  - To hold our parallel machines together, at the core of parallel computer architecture
  - Share basic concept with LAN/WAN, but very different trade-offs due to very different time scale/requirements
- Interconnection networks can be grouped into four domains[分类]
  - Depending on number and proximity of devices to be connected



Different Scales of Networks

- **Local-Area Networks**[局域网络]
  - Interconnect autonomous computer systems
  - Machine room or throughout a building or campus
  - Hundreds of devices interconnected (1,000s with bridging)
  - Maximum interconnect distance
    - Few meters to tens of kilometers
    - Example (most popular): Ethernet, with 10 Gbps over 40Km
- **Wide-Area Networks**[广域网络]
  - Interconnect systems distributed across the globe
  - Internetworking support is required
  - Millions of devices interconnected
  - Maximum interconnect distance
    - many thousands of kilometers



Different Scales of Networks (cont.)

- **System-Area Networks**[系统区域网络]
  - Interconnects within one “machine”
    - Interconnect in a multi-processor system
    - Interconnect in a supercomputer
- Hundreds to thousands of devices interconnected
  - Tianhe-2 supercomputer (16K nodes, each with 2 12-core processors)
- Maximum interconnect distance
  - Fraction to tens of meters (typical)
  - A few hundred meters (some)
    - InfiniBand: 120 Gbps over a distance of 300m

Different Scales of Networks (cont.)

- **On-Chip Networks**[片上网络]
  - Interconnect within a single chip
- Devices are micro-architectural elements
  - Caches, directories, processor cores
- Currently, designs with 10s of devices are common
  - Ex: IBM Cell, Intel multicores, Tile processors
- Projected systems with 100s of devices on the horizon
- Proximity: millimeters

We are concerned with On-Chip and System-Area Networks