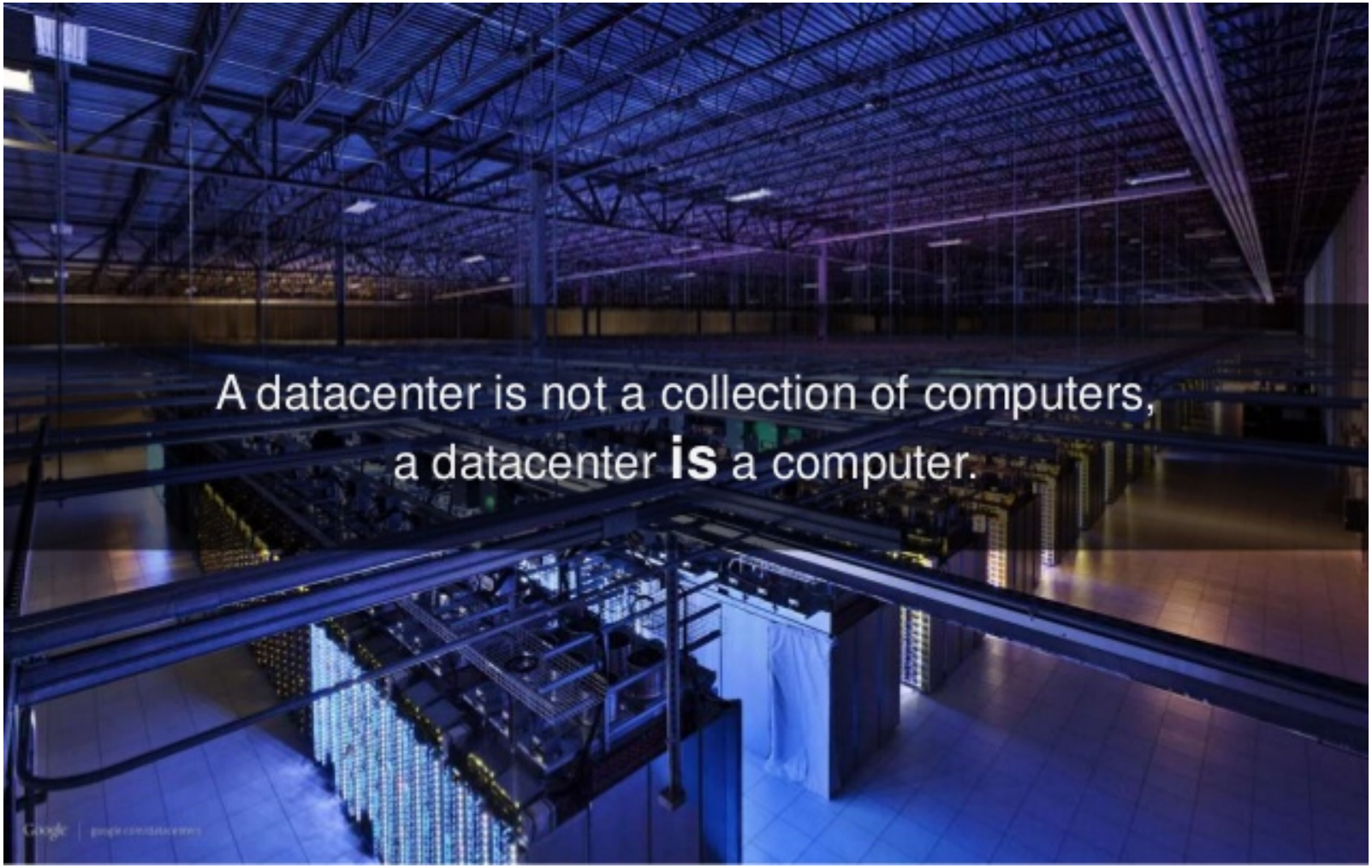
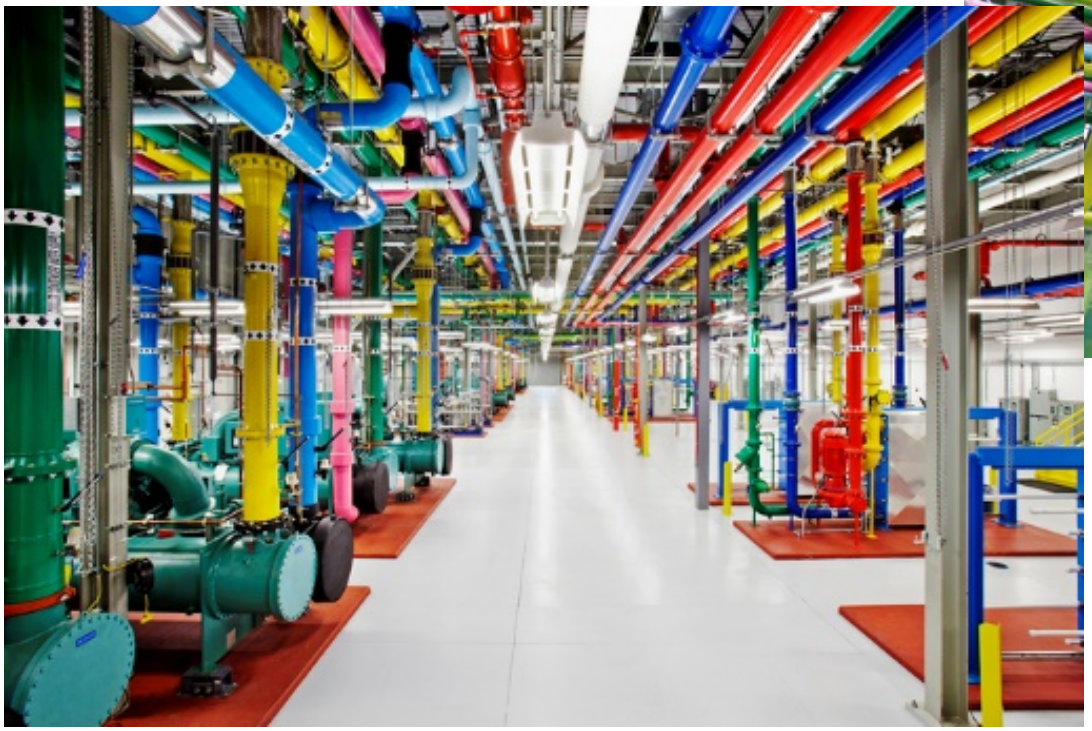
Handout 7
Warehouse Scale Computers
to Exploit Request-Level and Data-Level Parallelism

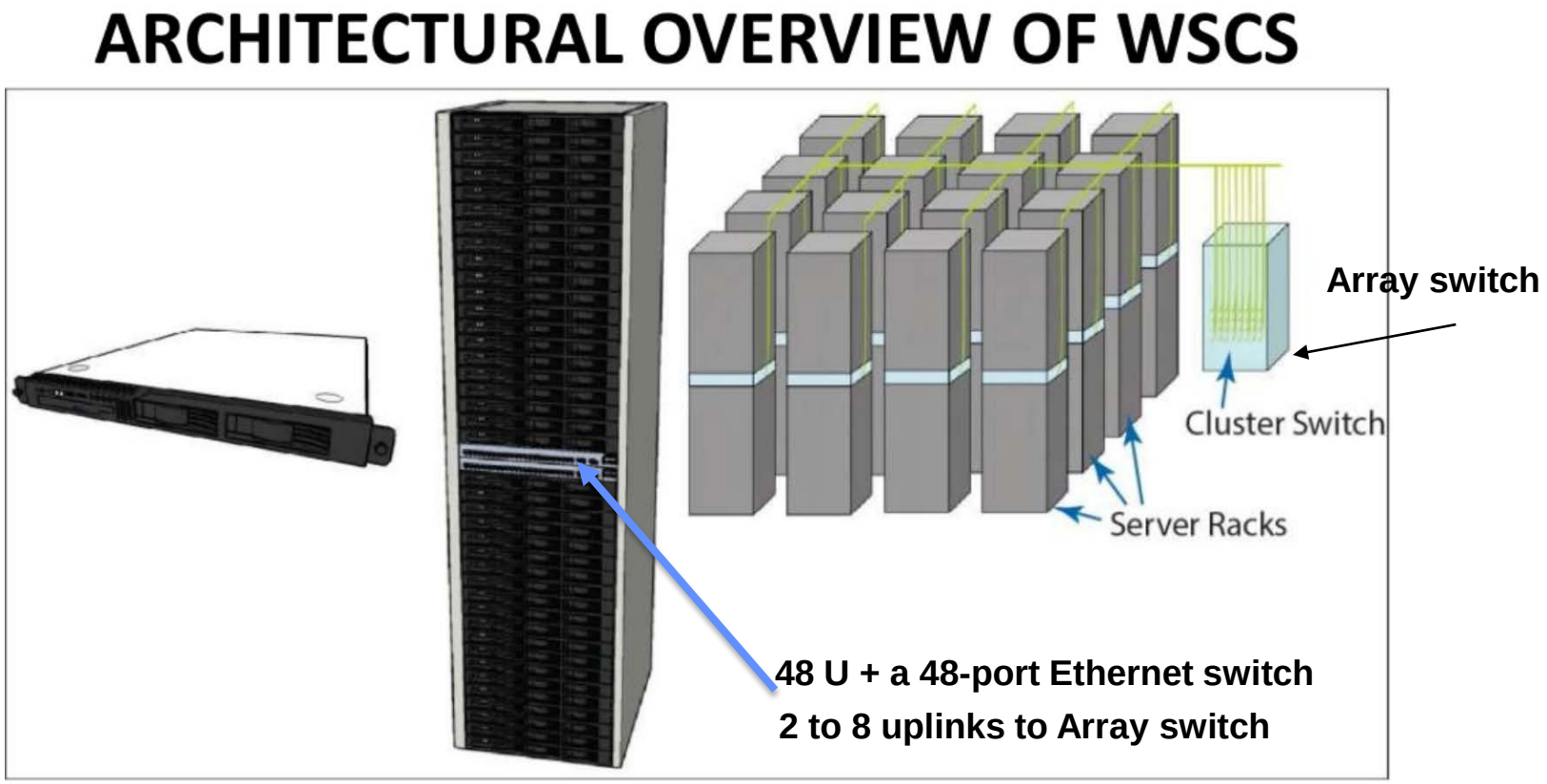
Data Center built upon WSC



Inside a Google Datacenter



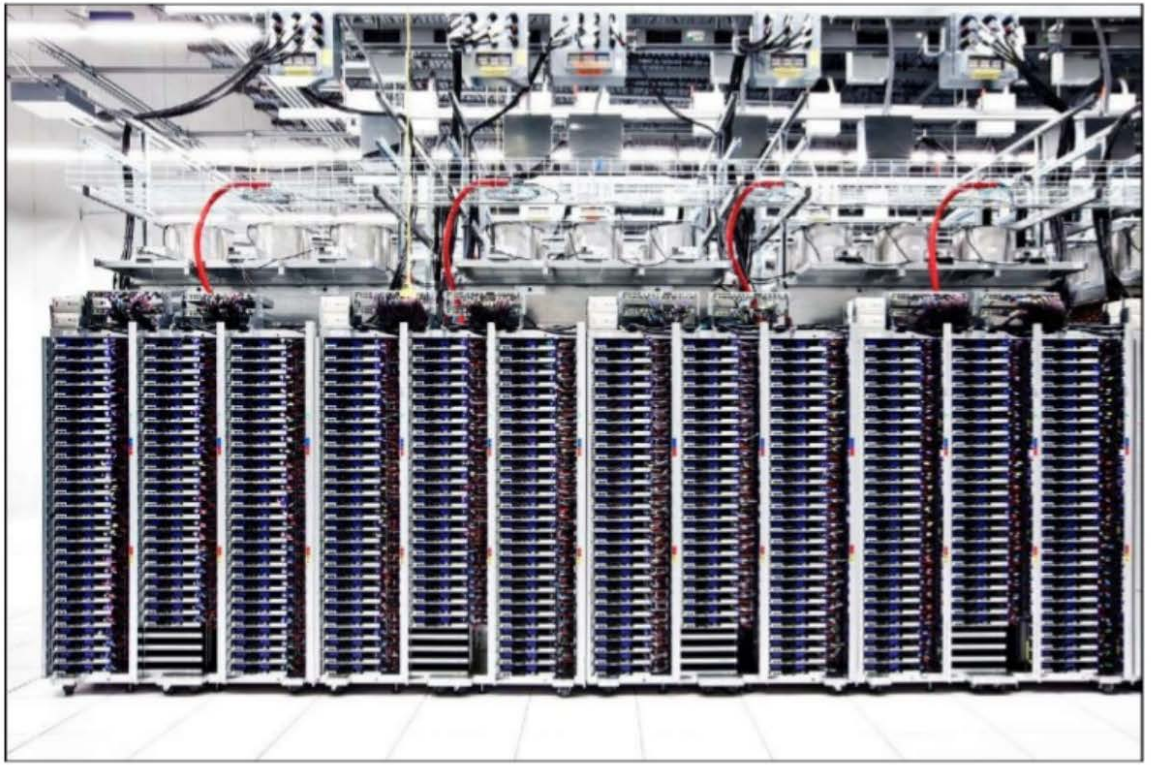
Warehouse Scale Computers



Sketch of the typical elements in warehouse-scale systems: 1U server (left), 7' rack with Ethernet switch (middle), and diagram of a small cluster with a cluster-level Ethernet switch/ router (right).

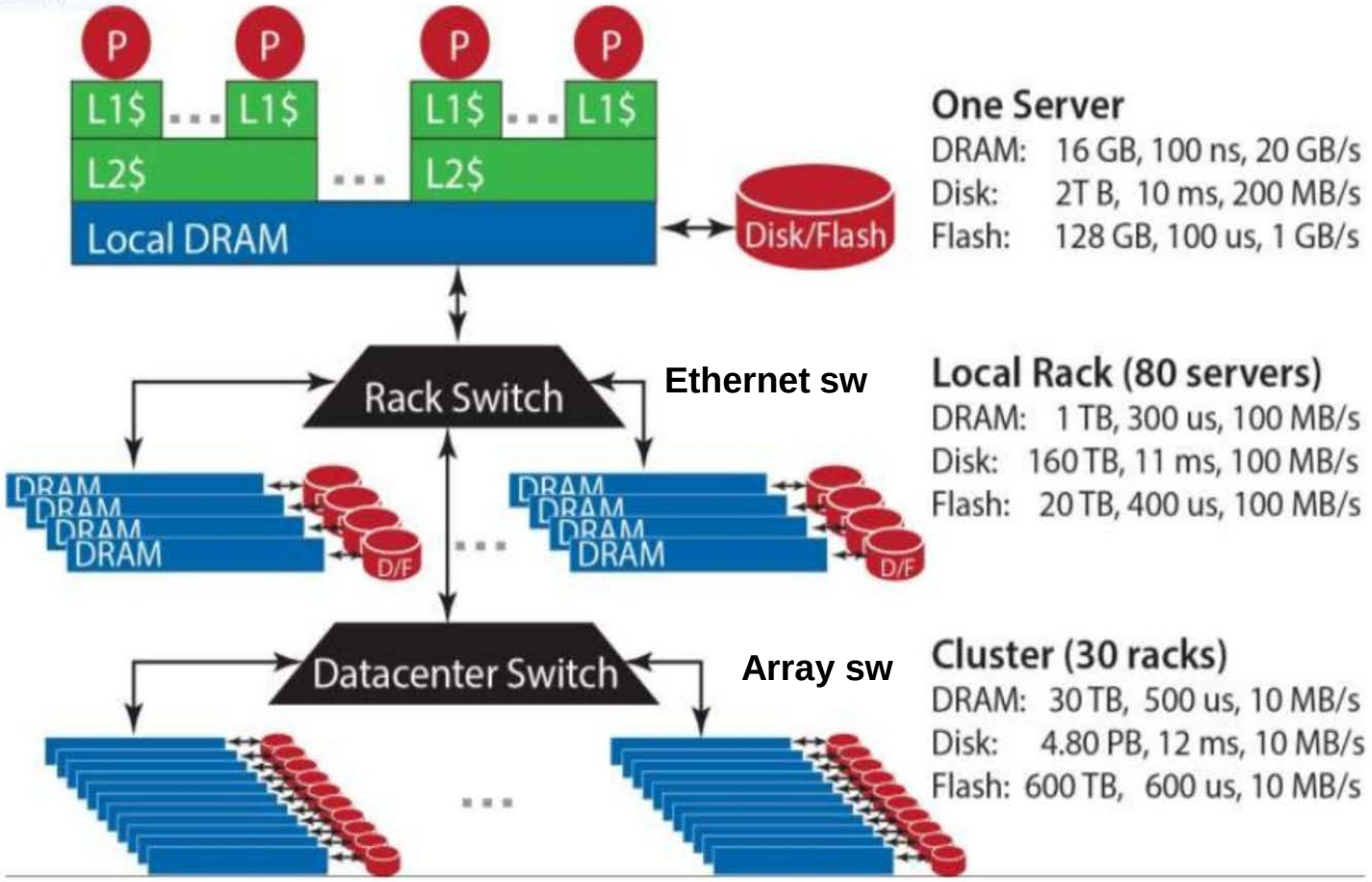
Servers in a rack

ARCHITECTURAL OVERVIEW OF WSCS



Picture of a row of servers in a Google WSC, 2012.

Switch View


P
L1$
L2$
Local DRAM
Disk/Flash
One Server
DRAM: 16 GB, 100 ns, 20 GB/s
Disk: 2T B, 10 ms, 200 MB/s
Flash: 128 GB, 100 us, 1 GB/s
Rack Switch
Ethernet sw
DRAM
D/F
Local Rack (80 servers)
DRAM: 1 TB, 300 us, 100 MB/s
Disk: 160 TB, 11 ms, 100 MB/s
Flash: 20 TB, 400 us, 100 MB/s
Datacenter Switch
Array sw
Cluster (30 racks)
DRAM: 30 TB, 500 us, 10 MB/s
Disk: 4.80 PB, 12 ms, 10 MB/s
Flash: 600 TB, 600 us, 10 MB/s

Memory Hierarchy (NUMA)

- **WSC Memory Hierarchy**

Example

	Local	Rack	Array
DRAM latency (microseconds)	0.1	100	300
Disk latency (microseconds)	10,000	11,000	12,000
DRAM bandwidth (MB/sec)	20,000	100	10
Disk bandwidth (MB/sec)	200	100	10
DRAM capacity (GB)	16	1,040	31,200
Disk capacity (GB)	2000	160,000	4,800,000

Each server contains:
16 GBytes of memory with a 100-nanosecond access time and transfers at 20 GBytes/sec and
2 terabytes of disk that offers a 10-millisecond access time and transfers at 200 MBytes/sec.

There are two sockets per board, and they share one 1 Gbit/sec Ethernet port.

Memory Hierarchy

- **WSC Memory Hierarchy**

Example

	Local	Rack	Array
DRAM latency (microseconds)	0.1	100	300
Disk latency (microseconds)	10,000	11,000	12,000
DRAM bandwidth (MB/sec)	20,000	100	10
Disk bandwidth (MB/sec)	200	100	10
DRAM capacity (GB)	16	1,040	31,200
Disk capacity (GB)	2000	160,000	4,800,000

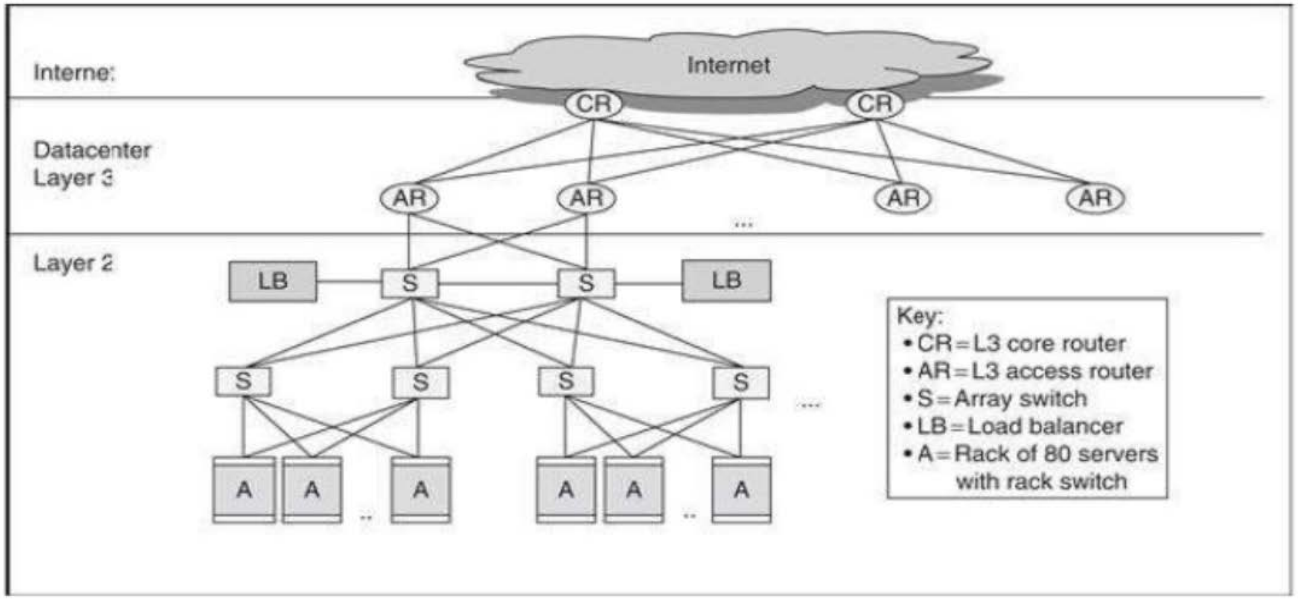
Every pair of racks includes one rack switch and holds 80 2U servers.

Networking software plus switch overhead increases the latency to DRAM to 100 microseconds and the disk access latency to 11 milliseconds.

Thus, the total storage capacity of a rack is roughly 1 terabyte of DRAM and 160 terabytes of disk storage.

The 1 Gbit/sec Ethernet limits the remote bandwidth to DRAM or disk within the rack to 100 MBytes/sec.

Switch Hierarchy and Network



Border router

L3 data center switch

Array switch

Rack switch

The Layer 3 network used to link arrays together and to the Internet [Greenberg et al. 2009].

Some WSCs use a separate *border router to connect the Internet to the datacenter Layer 3 switches.*

Computer Architecture of WSC

- WSC often uses a hierarchy of networks for interconnection
- Each 19” rack holds 48 1U servers connected to a rack switch
- Rack switches are uplinked to switch higher in hierarchy
  - Uplink has 48 / n times lower bandwidth, where n = # of uplink ports
    - » “Oversubscription”, is the ratio, for instance 24,
      - A 48-port Ethernet switch and 2 uplinks: 48/2 = 24; if 8 uplinks, 48/8 = 6
    - » A large “oversubscription” means that uplink bandwidth is much smaller than intra-rack bandwidth
  - Goal is to maximize locality of communication relative to the rack

Storage

- **Storage options:**
  - **Use disks inside the servers, or**
  - **Network attached storage through Infiniband (switched fabric)**
  - **WSCs generally rely on local disks**

Infiniband (commonly used in supercomputer)

- **InfiniBand uses a switched fabric topology**
  - **50 Gbit/s, one link. (x4, x8, x 12)**
- **InfiniBand transmits data in packets of up to 4 KB that are taken together to form a message. A message can be:**
  - **a direct memory access read from or write to a remote node (RDMA)**
  - **a channel send or receive**
  - **a transaction-based operation**
  - **a multicast transmission**
  - **an atomic operation**

Array Switch

- Switch that connects an array of racks
  - Array switch should have 10 X the bisection bandwidth of rack switch
  - Cost of *n*-port switch grows as n^2
  - Often utilize content addressible memory chips and FPGAs

Storage Consistency Model

Weak Consistency

- The protocol is said to support weak consistency if:
- **All accesses to synchronization variables** are seen by all processes (or nodes, processors) in the same order (sequentially) - these are synchronization operations.
- Accesses to critical sections are seen sequentially.
- All other accesses may be seen in different order on different processes (or nodes, processors).
- The set of both read and write operations in between different synchronization operations is the same in each process.

Strong Consistency

- The protocol is said to support strong consistency if:
- **All accesses** are seen by all parallel processes (or nodes, processors etc.) in the same order (sequentially)
- Therefore only one consistent state can be observed, as opposed to weak consistency, where different parallel processes (or nodes etc.) can perceive variables in different states.

Warehouse Scale Computers

- **Warehouse-scale computer (WSC)**
- **Provides Internet services**
- **Search, social networking, online maps, video sharing, online shopping, email, cloud computing, etc.**
- **Use a relatively homogeneous hardware and system software platform**
- **Share a common system management layer: in-house built AP, middleware, and system software**
- **Differences with HPC "clusters":**
  - **Clusters have higher performance processors and network**
- **Clusters emphasize thread-level parallelism, WSCs emphasize request-level parallelism**
- **Differences with traditional datacenters:**
  - **Datacenters consolidate different machines and software into one location**
  - **Datacenters emphasize virtual machines and hardware heterogeneity in order to serve varied customers**

WSC Design Factor

- Warehouse-scale computer (WSC)
  - Provides Internet services
    - » Search, social networking, online maps, video sharing, online shopping, email, cloud computing, etc.
  - Houses 50,000 to 100,000 servers
  - Design for
    - » Scale
    - » Dependability
    - » Debug ability

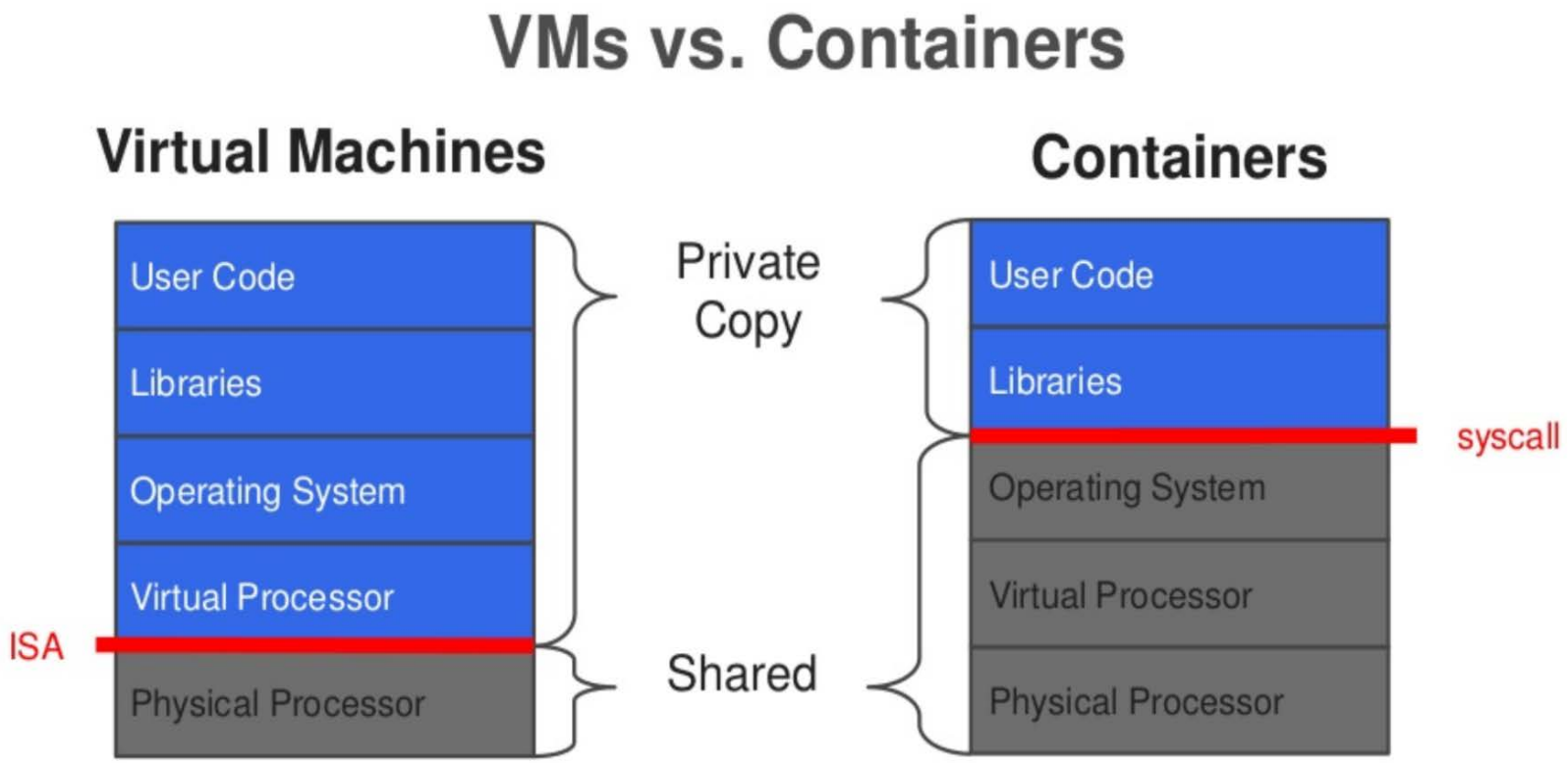
Google data centers process an average of 40 million searches per second

http://www.datacenterknowledge.com/archives/2017/03/16/google-data-center-faq

WSC in Container Form

- **A Google Warehouse-Scale Computer**
- **Containers**
- Both Google and Microsoft have built WSCs using shipping containers.
- The idea of building a WSC from containers is to make WSC design modular.
- Each container is independent, and the only external connections are **networking, power, and water**.
- The containers in turn supply networking, power, and cooling to the servers placed inside them, so the job of the WSC is to supply networking, power, and cold water to the containers and to pump the resulting warm water to external cooling towers and chillers.

Another Container (Docker)



Design Factors (1)

- **Important design factors for WSC:**
  - **Cost-performance**
    - » **Small savings add up**
  - **Energy efficiency**
    - » **Affects power distribution and cooling**
    - » **Work done per joule**
  - **Dependability via redundancy (Availability, 99.99%)**
    - » **Down time < 1 hr/year**
    - » **Redundancy management (multi-WSC)**
  - **Network I/O**
  - **Both interactive (like search) and batch processing workloads (parallel batch programs to compute metadata useful to search, for instance)**

Design Factor (2)

- **Important design factors for WSC:**
  - **Ample computational parallelism is not important**
    - » **Interactive Internet services: software as a service (SaaS)**
    - » **Most jobs are totally independent**
    - » **“Request-level parallelism”**
  - **Operational costs count**
    - » **Power consumption is a primary, not secondary, constraint when designing system**
  - **Scale and its opportunities and problems**
    - » **Can afford to build customized systems since WSC require volume purchase**
    - » **Expect one disk failure per hour for 50,000 severs WSC**

Get results from many servers

- **MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. (Google up to 2013)**
- **Hadoop (open-source)**
  - **Facebook runs Hadoop using 2000 batch processing servers out of 60,000 severs used in 2011**

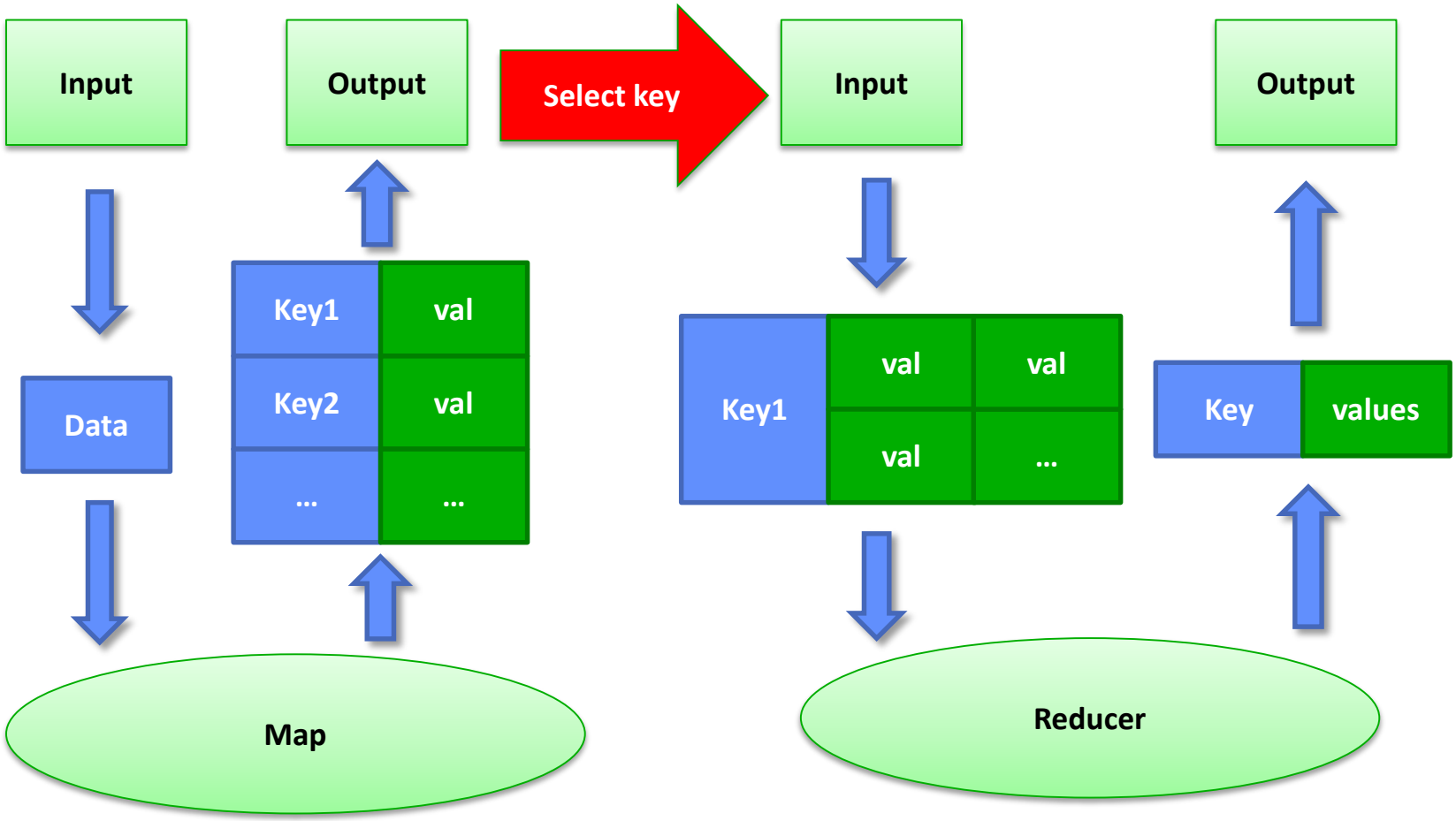
Map and Reduce

- A MapReduce program is composed of a **Map()** procedure (method) that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name)
- Input form: the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types. For instance, map(String key, String value) where key is document name, value is document contents.
- A **Reduce()** method that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies). For instance, reduce (String key, Iterator values)

Programming Models and Workloads

- **MapReduce runtime schedules the map tasks and the reduce task to the nodes of a WSC.**
  - **Map: applies a programmer-supplied function to each logical input record**
    - » **Runs on thousands of computers**
    - » **Provides new set of key-value pairs as intermediate values**
  - **Reduce: collapses values using another programmer-supplied function**
- **Analogy: MapReduce == SIMD : pass a function to data then a function in reduction of the output of the Map task**

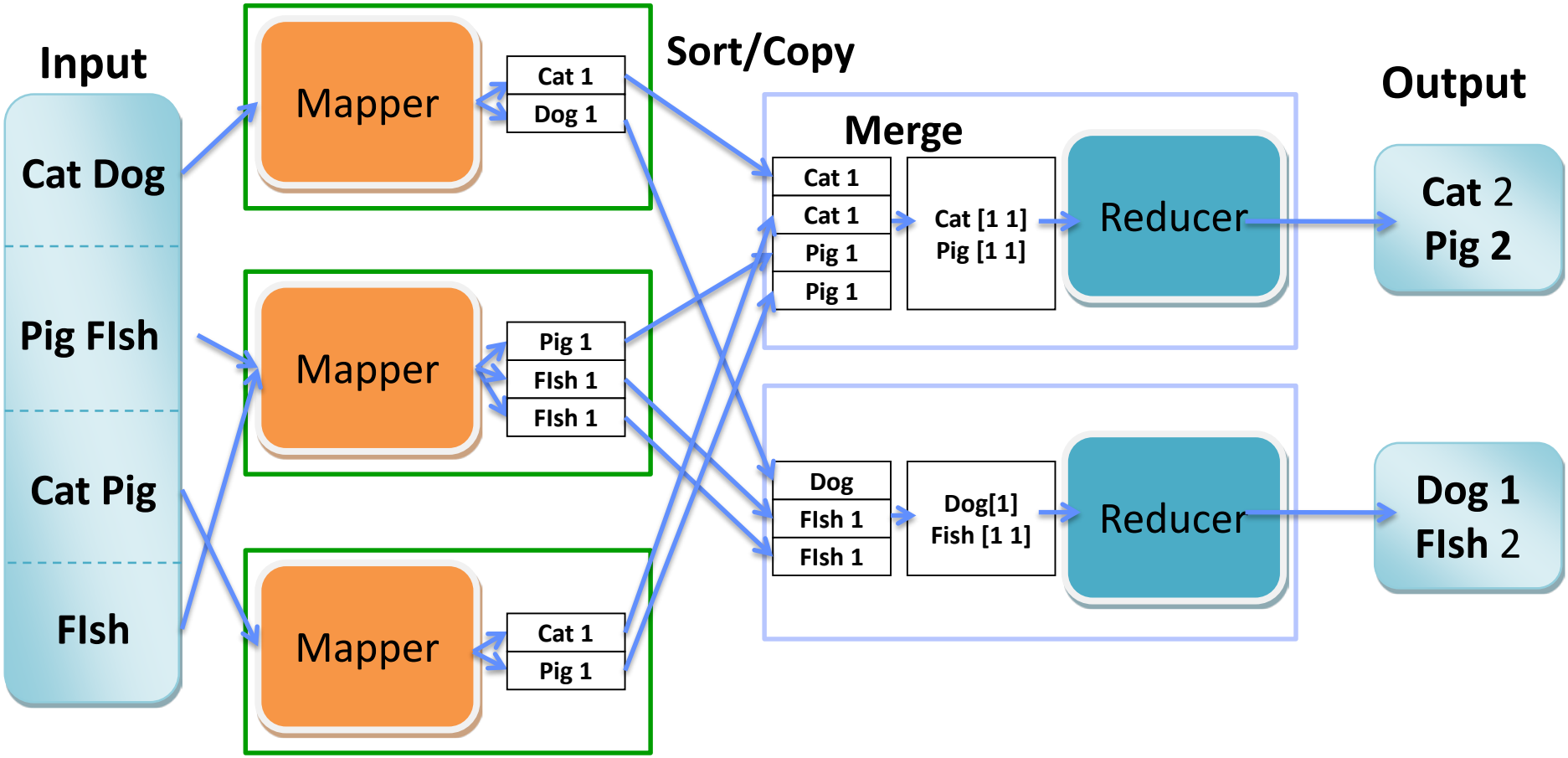
MapReduce and <Key, Value>



Run on many computers with each seeing different data segment, thus SIMD. Single Function to Perform.

Wordcount example

- Input can come from different web sites. (web pages = index files)



Application of MapReduce

- Text tokenization
  - Tokenization is the process of converting a sequence of characters (such as in a computer program or web page) into a sequence of tokens (strings with an assigned and thus identified meaning)
- Indexing and Search
- Data mining
- Machine learning

MapReduce Scheduler

- Schedule a function to thousands of computers (like SIMD)
- Task response time of a node determines
  - How soon will the next task come?
- Software mechanism handles slow task since it can hold up the completion of a large MapReduce job.
  - For instance, take the results from whichever finishes first while start backup executions on other nodes for tasks that are not completed yet.

MapReduce Ecosystem

- MapReduce runtime environment schedules map and reduce task to WSC nodes,
- and rely on Google File System (GFS) to supply files to any computer and various storage systems
  - » Google File System (GFS) uses local disks and maintains at least three relicas (cross machines)
  - » New FS Colossus has replaced GFS
- Want more for Availability:
  - Use replicas of data across different servers
  - Use relaxed consistency:
    - » No need for all replicas to always agree
    - » Hope to agree eventually

Measuring Efficiency of a WSC

- **Power Utilization Effectiveness (PUE)**
  - **PUE = Total facility power / IT equipment power**
  - **Median PUE on 2006 study was 1.69**
  - **Power usage effectiveness (PUE) is a ratio of how efficiently a computer data center uses energy; specifically, how much energy is used by the computing equipment (in contrast to cooling and other overhead).**
  - **PUE is the ratio of total amount of energy used by a computer data center facility to the energy delivered to computing equipment.**
  - **PUE was developed by a consortium called The Green Grid. An ideal PUE is 1.0. Anything that isn't considered a computing device in a data center (i.e. lighting, cooling, etc.) falls into the category of facility energy consumption.**

Measuring Efficiency of a WSC

- Performance
  - Latency is important metric because it is seen by users
  - Bing study: users will use search less as response time increases
  - Service Level Objectives (SLOs)/Service Level Agreements (SLAs)
    - E.g., 99% of requests be below 100 ms

DL in place for better PUE

- In July 2016, Google announced results from a test of an AI system by its British acquisition DeepMind. That system had reduced the energy consumption of its data center cooling units by as much as 40% and overall PUE by 15%. The system predicts temperatures one hour in advance, allowing cooling to be adjusted in anticipation.