Chapter 6

Warehouse-Scale Computers

to

Exploit Request-Level

and

Data-Level Parallelism

Warehouse-Scale Computers: Introduction

- Warehouse-scale computer (WSC)
  - Provides Internet services
    - Search, social networking, online maps, video sharing, online shopping, email, collaborative editing/design, cloud computing, etc.
  - Differences with HPC clusters:
    - Clusters have higher performance processors and network
      - WSC focus on commodity hardware
    - Clusters emphasize thread-level parallelism, WSCs emphasize request-level parallelism
  - Differences with datacenters:
    - Datacenters consolidate different machines and software into one location
      - WSC present a unified software model
    - Datacenters emphasize virtual machines and hardware heterogeneity in order to serve varied customers
      - WSC emphasize homogeneity

Warehouse-Scale Computers: Design Factors

- Cost-performance: small savings add up at scale
- Energy efficiency
  - Affects power distribution and cooling
  - Work per joule has to be maximized due to scaling
- Dependability via redundancy (99.99% availability at minimum)
- Network I/O (the one that is external to WSC)
- Interactive (search, social) and batch processing (index) workloads

Just like server architecture

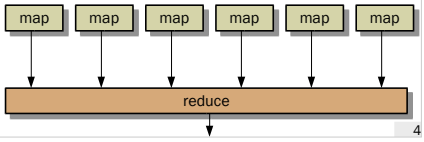
- Ample computational parallelism is not important
  - Most jobs are independent: request-level parallelism (SaaS, Web crawl)
- Operational costs count
  - Power consumption is a primary, not secondary, constraint when designing system (30% of operational cost in 10 years)
- Scale and its opportunities and problems
  - Can afford to build customized systems since WSC require volume purchase

Unlike server architecture

Prevalent Programming Model: MapReduce

- Implementations: MapReduce (Google), Hadoop (Apache)
- Sample code

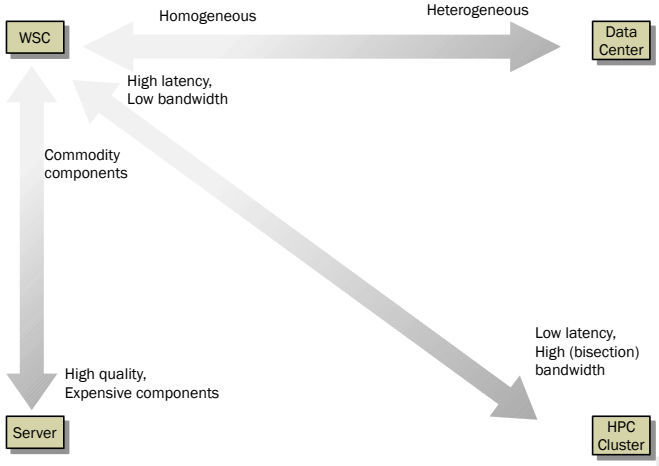
```
# applies a programmer-supplied function to each logical input
# record
def map(key : string, value : string) → pair:
    for item in value:
        EmitIntermediate(...) # produce key-value pairs

# collapses values using another programmer-supplied function
def reduce(key : string, values : iterator):
    for item in values:
        result = f(item)
        Emit( string(result) )
```



MapReduce Execution

- MapReduce runtime manages MapReduce jobs
  - Assigns MAP tasks to nodes based on how fast the nodes execute
    - This balances the load
  - Replicates execution of tasks and lets the nodes race
    - Improves completion time but (somewhat) waists resources
  - Counteracts hardware failures by rerunning failed tasks
  - Provides stable storage
    - Sample implementation: Google File System, Dynamo (Amazon), Big Table
  - Replicates data
    - Using erasure coding for more efficient storage
    - Improves resilience and performance (multiplies bandwidth by number of replicas)
  - Delivers storage consistency
    - Standard model (from databases): ACID (atomicity, consistency, isolation, durability) cannot be maintained (or is too costly) at WSC scale
  - Deals with variability in utilization (as high as 100% change)
- Do you need reliable hardware if software has built-in reliability?

WSC vs Data Center vs HPC Cluster vs Server



Hardware Architecture of WSC

- Node (server) count ~ 50k
- Connected with hierarchy of networks to reduce per-port cost
- Nodes (servers) are held in 19" by 7' racks that hold 48 units = 48U
- Commodity GigE switch offers 48 ports to accommodate the standard rack
  - The uplink count varies (2-8) which gives oversubscription in terms of bandwidth (48/2 to 48/8)
    - Software scheduler should aim at mapping sender and receiver to the same rack
      - Principle of locality
- Storage provided by local disks attached to a server inside the rack
  - External access through the Ethernet switches
  - NAS storage is too expensive per TB
    - It has features not needed in WSC, for example, RAID not needed due to resilience in software

WSC Networking: Array Switch

- Connects arrays of racks with each other
- Standard 48-port Ethernet switch is insufficient
  - Oversubscription problem
  - Too low internal bandwidth
- Key performance metric: bisection bandwidth
  - 10 times higher than the standard 48-port switch
  - Calculation of bisection bandwidth
    - Take the worst bandwidth out of…
      - All partitions of the ports into two disjoint groups…
        - Communicating at the same time
  - Worst case scenario under workload
- Cost: 100 times the standard Ethernet switch, because…
  - Higher bisection bandwidth (cost of n ports grows as n^2)
  - Higher profit margins due to cost of parts (FPGAs, …), lower volume
  - Extra features: high rate DPI (Deep Packet Inspection)

WSC Memory and Storage Hierarchy

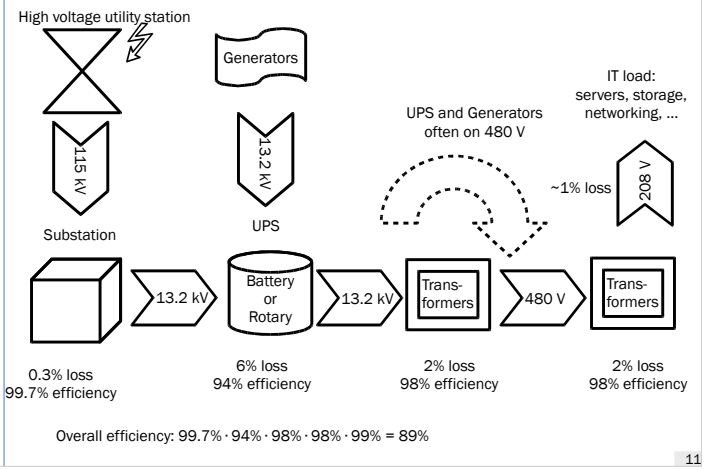
		Local	Rack	Array
DRAM latency	microseconds	0.1	100	300
Disk latency	microseconds	10 000	11 000	12 000
DRAM bandwidth	MB/s	20 000	100	10
Disk bandwidth	MB/s	200	100	10
DRAM capacity	GB	16	1040	31 200
Disk capacity	GB	2000	160 000	4 800 000

- Every pair of racks includes one rack switch and holds 80 2U servers
- The network makes remote disk and DRAM work equally fast
- Most applications fit within a single array
  - When more storage is required, use sharding or partitioning
- Array switches may be stacked in a multi-level hierarchy
  - Load balancers are at the highest level

WSC Physical Infrastructure Costs

- Location considerations – proximity to…
  - Internet backbone optical fibers
  - Low cost electricity
  - Low risk of environment disasters (earthquakes, floods, hurricanes, …)
  - Geographical vicinity of large population of users
  - Real estate deals and low property taxes
- Construction cost is negligible
- Power distribution and cooling are major cost considerations

Electrical Power Distribution



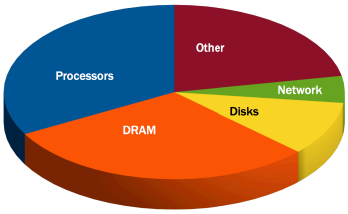
Overall efficiency: 99.7% · 94% · 98% · 98% · 99% = 89%

Cooling Infrastructure & Computer Room A/C

- Most common air cooling systems use phase change design
  - Components: condenser, compressor, evaporator, pumps, fans
  - Cool air for servers: 64°F – 71°F
  - Energy efficiency increases when temperature is allowed to go up
    - But higher temperature increases failure rate as well as the wear of components
  - Outside air can also be used if it is cold enough
    - Use wet-bulb test to find out minimum temperature achieved with evaporating water with air
  - Careful separation of cold and hot air helps with energy efficiency
    - Do not allow front of the server to face the back of another server
  - Power budgets
    - Chillers: 30%-50% of IT power
    - CRAC: 10%-20% of IT power
  - Server power draw varies depending on the load
    - As much as 40% oversubscription is possible beyond the estimate
      - Software de-schedule lower-priority tasks in case of power over draft

Cooling and Power Draw

- Cooling systems circulate, evaporate and spill water
  - An 8MW facility: 70 – 200 thousand gallons of water per day
- Breakdown of power budget in 2007
  - 33% processors
  - 30% DRAM
  - 10% disks
  - 5% networking
  - 22% other



Measuring Efficiency of a WSC

- Commonly used metric is Power Utilization Effectiveness
  - PUE must be greater than 1 but closer to 1 is better
  - Back in 2006: PUE = 1.33 ... 3.03 (median 1.69)
  - Currently: breaking the 1.05 barrier (NREL supercomputing facility)
  - PUE tricks: where is the power measured, what is the workload, ...

$$\text{PUE} = \frac{\text{Total facility power}}{\text{IT equipement power}}$$

- User-level efficiency metrics
  - Latency (of request, first response, completion...) is the immediately perceivable metric from users' perspective
  - User productivity = 1 / time of interaction
    - t(interaction)=t(human entry)+t(system response)+t(analysis of response)
  - From experiments: t(response) down by 30% ⟶ 70% less t(interaction)
    - Because people think continuously when not interrupted by a long delay
  - Bing experiment:
    - 200 ms longer delay on server ⟶ 500 ms longer time to next click
    - Revenue and user satisfaction drops linearly with increasing delay
  - Google experiment: effects of delays linger

Primary Concern: User Satisfaction

- Based on Internet studies...
  - Page load above few tens of milliseconds cause user to switch to another task
  - Page load time must be below 1s or it's deemed broken
    - Users do not come back
- Quantifying influence of response delays
  - SLO = Server Level Objective
  - SLA = Server Level Agreement
    - Example: 99% of requests must be below 100 ms delay
    - Amazon's Dynamo: 99.9% of key-value request must be below threshold
      - Which is more important average case of the tail (diminishing return)?

Cost of a WSC

- Operational Expenditures = OPEX
- Capital Expenditures = CAPEX
- US accounting rules allow to extract OPEX value from CAPEX value
  - Must use amortization and average life time of components
  - The cost calculation may get complicated for long term investments, eventual upgrades, facility expansion, etc.
  - Sample calculation (2010)
    - Amortized CAPEX
      - Servers: 53%
      - Network hardware: 8%
      - Power and cooling: 20%
      - Other: 4%
    - OPEX
      - Monthly power use: 13%
      - Monthly people: 2%
- Estimates of CAPEX and OPEX give an idea of where to invest to cut costs: software, hardware, infrastructure

Cloud Computing

- Cloud computing enabled by CAPEX/OPEX at the user level, data center, and WSC scale (economies of scale)
  - It might make economic sense to migrate from in-house to WSC
    - Better negotiated price for volume purchases
    - Faster delivery for large purchases
      - On-demand growth
    - Uniformity of hardware and software helps with administration
    - Better server utilization (increase from 10%-20% to as much as 50%)
  - Other costs/discounts not included directly:
    - Cost of data (acquisition and loss)
    - Cost of privacy and accounting rules (Sarben-Oxley, HIPPA, ...)
    - Competitive advantage: time-to-market
  - Cost reduction by going from data center to WSC in 2010
    - Storage: 5.7x
    - Administration: 7.1x
    - Networking: 7.3x

Example: Cloud Computing with Amazon

- Services provided: S3, AWS, Dynamo, Glacier, ...
- Based on business decisions:
  - Virtualization
    - True time sharing and user-to-user protection
    - Software distribution is simplified
    - Management of software life time: migration, balancing,...
    - Control over resource use that gives potential for variable pricing
    - Decoupling of hardware and software for seamless upgrades
      - Hardware platform is a VM which defines a generic x86 machine
  - Low-cost (competitive) hourly rate
  - (Initial) reliance on Open Source Software
    - Later on, commercial software companies had to adapt
  - (Initial) lack of guarantee of service
    - Evolved into SLA levels
  - No contract required
  - No danger of over- or under-provisioning

AWS and Cloud Computing Poster Children

- Zynga's FarmVille: case for growth-on-demand
  - After 4 days: 1 million players
  - After 60 days: 10 million players
  - After 270 days: 28 million players
- Netflix
  - On-demand encoding for evolving screen sizes and form factors
    - Optimization of bandwidth use by choosing low-res video stream
    - Long decode times can be easily distributed
    - Adding a new content partner means a surge of decode/encode demand
  - On-demand networking through CDN's: 22% to 30% of Internet traffic
  - Competitive issues: Netflix vs. Amazon streaming