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

Introduction

- 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

Introduction-1

- **Important design factors for WSC: 1/2**
  - **Cost-performance**
    - » **Small savings add up**
  - **Energy efficiency**
    - » **Affects power distribution and cooling**
    - » **Work done per joule**
  - **Dependability via redundancy (Availability, 99.99%)**
    - » **down < 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)**

Introduction-2

- Important design factors for WSC: 2/2
  - 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

Introduction-3

- **Warehouse-scale computer (WSC)**
  - **Differences with HPC “clusters”:**
    - » Clusters have higher performance processors and network
    - » Clusters emphasize thread-level parallelism, WSCs emphasize request-level parallelism
  - **Differences with datacenters:**
    - » Datacenters consolidate different machines and software into one location
    - » Datacenters emphasize virtual machines and hardware heterogeneity in order to serve varied customers
    - » WSC means to act like a single computer that runs a variety of applications

Get results from many servers

- MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. (Google)
- Hadoop (open-source)
  - Facebook runs Hadoop using 2000 batch processing servers out of 60,000 severs used in 2011
- Map runs the same function to each logical input record on lots of computers and gets an intermediate result of Key-Value pair.
- Reduce collects these results from each of the distributed tasks and collapse them using a programmer specified function

Operations

- Functional Programming : Map and Reduce
- –map(...) :
  - Input: RAW data
  - Output: (Key1, value)(Key2, value …)
- –reduce(...):
  - Input: (Key1, val, val, val..); select a key
  - Output: (Key1: values)
- Divide and conquer

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**

Application of MapReduce

- Text tokenization
- 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
- 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

Computer Architecture of WSC

- WSC often use 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**
  - **Google File System (GFS) uses local disks and maintains at least three relicas**

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