Operating Systems and The Cloud

David E. Culler
CS162 – Operating Systems and Systems Programming
Lecture 39
December 1, 2014
Goals Today

• Give you a sense of kind of operating systems issues that arise in The Cloud

• Encourage you to think about graduate studies and creating what is out beyond what you see around you ...
The Datacenter is the new Computer ??

• “The datacenter as a computer” is still young
  – Complete systems as building blocks (PC+Unix+HTTP+SQL+ …)
  – Higher Level Systems formed as Clusters, e.g., Hadoop cluster
  – Scale ⇒ More reliable than its components
  – Innovation ⇒ Rapid (ease of) development, Predictable Behavior despite variations in demand, etc.
Datacenter/Cloud Computing OS ???

• If the datacenter/cloud is the new computer,
  • what is its **Operating System**?
    – Not the host OS for the individual nodes, but for the millions of nodes that form the ensemble of quasi-distributed resources!

• Will it be as much of an enabler as the LAMP stack was to the .com boom?

• Open source stack for every Web 2.0 company:
  – Linux OS
  – Apache web server
  – MySQL, MariaDB or MongoDB DBMS
  – PHP, Perl, or Python languages for dynamic web pages
Classical Operating Systems

- **Data sharing**
  - Inter-Process Communication, RPC, files, pipes, ...

- **Programming Abstractions**
  - Storage & I/O Resources, Libraries (libc), system calls, ...

- **Multiplexing of resources**
  - Scheduling, virtual memory, file allocation/protection, ...
Datacenter/Cloud Operating System

• Data sharing
  – Google File System, key/value stores
  – Apache project: Hadoop Distributed File System

• Programming Abstractions
  – Google MapReduce
  – Apache projects: Hadoop, Pig, Hive, Spark, …
  – Nyad, Driad, …

• Multiplexing of resources
  – Apache projects: Mesos, YARN (MapReduce v2), ZooKeeper, BookKeeper, …
Google Cloud Infrastructure

• Google File System (GFS), 2003
  – Distributed File System for entire cluster
  – Single namespace

• Google MapReduce (MR), 2004
  – Runs queries/jobs on data
  – Manages work distribution & fault-tolerance
  – Colocated with file system

• Apache open source versions: Hadoop DFS and Hadoop MR
GFS/HDFS Insights

• *Petabyte* storage
  – Files split into large blocks (128 MB) and replicated across many nodes
  – Big blocks allow high throughput sequential reads/writes

• Data *striped* on hundreds/thousands of servers
  – Scan 100 TB on 1 node @ 50 MB/s = 24 days
  – Scan on 1000-node cluster = 35 minutes

• *Failures* will be the norm
  – Mean time between failures for 1 node = 3 years
  – Mean time between failures for 1000 nodes = 1 day

• Use *commodity* hardware
  – Failures are the norm anyway, buy cheaper hardware

• No complicated consistency models
  – Single writer, append-only data
MapReduce Insights

• Restricted key-value model
  – Same fine-grained operation (Map & Reduce) repeated on huge, distributed (within DC) data
  – Operations must be deterministic
  – Operations must be idempotent/no side effects
  – Only communication is through the shuffle
  – Operation (Map & Reduce) output saved (on disk)
What is (was) MapReduce Used For?

- **At Google:**
  - Index building for Google Search
  - Article clustering for Google News
  - Statistical machine translation
  - ...

- **At Yahoo!**:
  - Index building for Yahoo! Search
  - Spam detection for Yahoo! Mail
  - ...

- **At Facebook**:
  - Data mining
  - Ad optimization
  - Spam detection
  - ...
A Time-Travel Perspective

3 Billion by …

1969
1974
1990
2010
WWW
Internet
ARPANet
RFC 675 TCP/IP

11/2004
2.0 B 1/26/11
2.8 B

Google

Data source: World Bank, World Development Indicators - Last updated December 21, 2010

United States
World
Research as “Time Travel”

• Imagine a technologically plausible future
• Create an approximation of that vision using technology that exists.
• Discover what is True in that world
  – Empirical experience
    » Bashing your head, stubbing your toe, reaching epiphany
  – Quantitative measurement and analysis
  – Analytics and Foundations
• Courage to ‘break trail’ and discipline to do the hard science
NOW – Scalable Internet Service Cluster Design

NOW Project Timeline

- ATM, fddi
- Start of Funding
- Myrinet
- 1/95
- 1/96
- 1/97
- 6/97
- 1/98
- 6/98

NOW 0
NOW I
NOW II
NOW Sort
NOW Finale

CS 252
CS 258
Inktomi
ASPSOS Workshop I
ASPSOS Workshop II
NPACI
CS 267
Google

VAX clusters

12/1/14
UCB CS162 Fa14 L39
1993 Massively Parallel Processor is King
NOW – Scalable High Performance Clusters

GSC+ => PCI => ePCI …

NOW Project Timeline

10m Ethernet, FDDI, ATM, Myrinet, … VIA, Fast Ethernet, => infiniband, gigEtherNet
NOW – Scalable High Performance Clusters
UltraSparc/Myrinet NOW

- Active Message: Ultra-fast user-level RPC
- When remote memory is closer than local disk ...
- Global Layer system built over local systems
  - Remote (parallel) execution, Scheduling, Uniform Naming
  - xFS – cluster-wide p2p file system
  - Network Virtual Memory
Inktomi – Fast Massive Web Search
Fiat Lux - High Dynamic Range Imaging

Paul Gauthier

NOW Project Timeline

ATM, fidi  Start of Funding  Myrinet  sgi  VIA  G-Ether  NOW Final

Paul Debevec

CS267, Spring 1995: Final Projects

- Fast Parallel Iterative Matrix Diagonalization
- Prolemy C Code Generation and Scheduling for the Network of Workstations (NOW)
- Parallel Raytracing using a Network of Workstations for Rendering Spline Surface Animation
- Parallel Monte Carlo Simulation
- Berkeley Search Engine
- Porting and Characterization of GATOR, an Atmospheric Chemical Tracer Model
- A Distributed Memory Concurrent B-tree Implementation
- Design, Implementation, and Performance Evaluation of a Portable Distributed Task Queue
- Porting The BLACS From MPI To GAM On The SP-1
- Implementation of a Preconditioned Conjugate Gradient (PCG) Solver in Finite Element
- Parallelizing Impulse, a dynamic simulation system
- Model of LPARX multigrid performance on the CM5

http://www.pauldebevec.com/FiatLux/movie/
inktomi.berkeley.edu

- World’s 1st Massive AND Fast search engine

1996 inktomi.com
Distributed File Storage stripped over all the disks with fast communication.
Massive Cheap Storage

Serving Fine Art at http://www.thinker.org/imagebase/
... google.com

NOW Project Timeline

N0 $’s in Search
Big $’s in caches
??? $’s in mobile

Yahoo moves from inktomi to Google
meanwhile Clusters of SMPs

Millennium Computational Community

- Gigabit Ethernet
- Business
- Chemistry
- Biology
- Astro
- Physics
- Economy
- Math
- Transport
- NERSC
- IEOR
- C.E.
- MSME
- C.S.
- E.
- M.E.
- N.E.
- BMRC
- SIMS
Expeditions to the 21st Century

Away from the ‘average’ Device

Info. appliances
Server
Client
Scalable, Available
Internet Services

8/16/99

Expeditions Diverse OS
Internet Services to support small mobile devices

Service Based Applications

- Application provides services to clients
- Grows/Shrinks according to demand, availability, and faults

UNTANGLING THE WEB: UC Berkeley graduate students Steve Gribble, Armando Fox, and Yatin Chawathe (left to right) have created a system called Transend that can speed up modem access to the World Wide Web by distilling image files.
Ninja Internet Service Architecture

Opportunity: infrastructure services

- Prehistoric: DNS, IP route tables, ...
- Historic: crawl, index, search,
- Emerging: compose and manipulate data and services

And client diversity has just begun!

6/4/2000  Java Grande
Example: Ninja Jukebox 98

- Collaborative Community: anyone can add content
  => mp3.com, real jukebox, napster
- Authentication and authorization was built-in
- Jukebox 99: Music similarity query engine
  => mongomusic.com, ...

- CD “ripper” service
- CDDB service

- Ninja iSpace
  - Fetched track/title & artist information from an online DB.

- Ninja iSpace
  - Web page with song playlists

- Music Directory service

- HTTPd service
Santio: universal instant messaging  S. Gribble

AOL client

AOL worker

english to spanish

ICQ client

ICQ worker

profile DDS

sanctio service (cluster)?
Existing Applications

- Ninja "NOW Jukebox"
  - Harnesses Berkeley Network of Workstations
  - Plays real-time MPEG-3 audio served from 11
- Voice-enabled room control
  - Speech-to-text Operators control room services
  - Eventual integration with GSM cell phones and
- Stock Trading Service
  - Accesses real-time stock data from Internet
  - Programmatic interface to buy/sell/trade stock
- NinjaFAX
  - Programmable remotely-accessed FAX machines
  - Send/receive FAXes; authentication used for a
- Keiretsu: The Ninja Pager Service
  - Provides instant messaging service via Web, In

Figure 2: Distributed hash table architecture: each box in the diagram represents a software process. In the simplest case, each process runs on its own physical machine, however there is nothing preventing processes from sharing machines.

Gribble, 99
Security & Privacy in a Pervasive Web

Composable, Secure Proxy Architecture for Post-PC devices

Diverse Clients

Personal Appl

Embedded Untrusted Client

Trusted Client

Filter and Control Modifier

Transient Store

Identity Service

Format Transcoders

Security Adapters

DATEK (Trust Contract)

Internet Services

Yahoo!

Bank of America

IRS

E-Trade

The Best Password Managers

BY NEIL J. RUBENKING    AUGUST 22, 2014    49 COMMENTS

In these days of hacks, Heartbleed, and endless breaches, a strong, unique, and often-changed password for every site is even more imperative. A password manager can help you attain that goal.

3.1K SHARES

LastPass 3.0

LastPass 3.0 Premium

Dashlane 3

RoboForm Everywhere 7

Intuitive Password 2.9

Keeper Password Manager & Digital Vault 8

Norton Identity Safe

PasswordBox

RoboForm Desktop 7

Sticky Password 7
A decade before the cloud

A ‘Structured Architecture’ Approach

- Bases (1M's)
  - scalable, highly available
  - persistent state
  - databases, agents
  - “home” base per user
  - service programming environment

- Active Proxies (100M's)
  - not packet routers
  - bootstrap thin devices into infrastructure
  - soft-state and well-connected

- Units (1B's)
  - sensors / actuators
  - PDAs / smartphones / PCs
  - heterogeneous
  - Minimal functionality: “Smart Clients”

99.9 Club

NOW Project Timeline

- ATM, fddi
- 1/94
- Start of Funding
- 1/95
- Myrinet
- 6/95
- VIA
- 6/96
- G-Ether
- 6/97
- NOW Finale
- 6/98
- Many PhDs
NOW Team 2008: L-R, front row: Prof. Tom Anderson†‡ (Washington), Prof. Rich Martin‡ (Rutgers), Prof. David Culler*†‡ (Berkeley), Prof. David Patterson*†‡ (Berkeley). Middle row: Eric Anderson (HP Labs), Prof. Mike Dahlin‡ (Texas), Prof. Armando Fox‡ (Berkeley), Drew Roselli (Microsoft), Prof. Andrea Arpaci-Dusseau‡ (Wisconsin), Lok Liu, Joe Hsu. Last row: Prof. Matt Welsh‡ (Google), Eric Fraser, Chad Yoshikawa, Prof. Eric Brewer*†‡ (Berkeley), Prof. Jeanna Neefe Matthews (Clarkson), Prof. Amin Vahdat‡ (UCSD), Prof. Remzi Arpaci-Dusseau (Wisconsin), Prof. Steve Lumetta (Illinois).

*3 NAE members †4 ACM fellows ‡9 NSF CAREER Awards
Time Travel

- It’s not just storing it, it’s what you do with the data

AMPLab Unification Philosophy

Don’t specialize MapReduce – Generalize it!

Two additions to Hadoop MR can enable all the models shown earlier!

1. General Task DAGs
2. Data Sharing

For Users:
- Fewer Systems to Use
- Less Data Movement

Making Sense of Big Data with Algorithms, Machines & People

Ion Stoica
EECS, Berkeley

Spark
Streaming
GraphX
MLlib
The Data Deluge

- Billions of users connected through the net
  - WWW, Facebook, twitter, cell phones, ...
  - 80% of the data on FB was produced last year

- Clock Rates stalled

- Storage getting cheaper
  - Store more data!
Data Grows Faster than Moore’s Law

Projected Growth

- Moore's Law
- Particle Accel.
- DNA Sequencers

Increase over 2010

2010 2011 2012 2013 2014 2015
Complex Questions

• Hard questions
  – What is the impact on traffic and home prices of building a new ramp?

• Detect real-time events
  – Is there a cyber attack going on?

• Open-ended questions
  – How many supernovae happened last year?
MapReduce Pros

• Distribution is completely transparent
  – Not a single line of distributed programming (ease, correctness)

• Automatic fault-tolerance
  – Determinism enables running failed tasks somewhere else again
  – Saved intermediate data enables just re-running failed reducers

• Automatic scaling
  – As operations as side-effect free, they can be distributed to any number of machines dynamically

• Automatic load-balancing
  – Move tasks and speculatively execute duplicate copies of slow tasks (stragglers)
MapReduce Cons

• Restricted programming model
  – Not always natural to express problems in this model
  – Low-level coding necessary
  – Little support for iterative jobs (lots of disk access)
  – High-latency (batch processing)

• Addressed by follow-up research and Apache projects
  – Pig and Hive for high-level coding
  – Spark for iterative and low-latency jobs
Complex jobs, interactive queries and online processing all need one thing that MR lacks:

Efficient primitives for data sharing

Iterative job
Interactive mining
Stream processing
Spark Motivation

Complex jobs, interactive queries and online processing all need one thing that MR lacks:

Efficient primitives for data sharing

Problem: in MR, the only way to share data across jobs is using stable storage (e.g. file system) → slow!
Examples

Opportunity: DRAM is getting cheaper → use main memory for intermediate results instead of disks
Goal: In-Memory Data Sharing

One-time processing

Input

Distributed memory

iter. 1

iter. 2

\[ \ldots \]

query 1

query 2

query 3

\[ \ldots \]

10-100\times \text{faster than network and disk}
Solution: Resilient Distributed Datasets (RDDs)

• Partitioned collections of records that can be stored in memory across the cluster

• Manipulated through a diverse set of transformations (map, filter, join, etc)

• Fault recovery without costly replication
  – Remember the series of transformations that built an RDD (its lineage) to recompute lost data

Berkeley Data Analytics Stack (open source software)

- Cancer Genomics, Energy Debugging, Smart Buildings
- BlinkDB
- Sample Clean
- MLBase
- SparkR
- Spark Streaming
- SparkSQL
- GraphX
- MLlib
- Apache Spark
- Velox (Model Serving)
- Tachyon
- HDFS, S3,
- Apache Mesos
- Yarn

In-house Apps
Access and Interfaces
Processing Engine
Storage
Resource Virtualization