Large Scale Computing
@ Linkedin

Bhupesh Bansal
Software Engineer
Linkedin
The Plan

• What Is Linkedin
• Large Scale problems @ Linkedin
• Scalable Architectures
• Hadoop : The mighty elephant
• Questions
What is LinkedIn?

- The largest **Professional** social network:

- Some high level statistics:
  - 50M active users, in 147 Industries and 244 Countries (580+ Regions)
  - ~10M unique daily visitors
  - ~25M weekly searches (150 QPS at peak)
  - ~50M weekly profile views
  - ~2M connections per day
Why should you care?

- Control your brand
- Market yourself
- Find inside connections
- Learn from Wisdom of the crowd
Build Your Brand
What is your Identity?
How do you want the world to see and know about you?
1. Explore opportunities before it's too late
2. Make your network go far. (well really really far !!)
3. Seek expert advice
4. Stand on the shoulder of giants.
Wisdom of your network

1. Slideshare: See slides uploaded by your network
2. Answers: See expert answers & who they are
3. Groups
4. ...
Large Scale Problems @ Linkedin
Linkedin Network Cache

- Third Degree network
  - Friends of Friends of Friends
  - Scales exponentially
    - first degree: 100
    - second-degree: first-degree^2
    - third-degree: second-degree^2
  - Ranges from 10,000 → 20 M.
  - Is Unique for every user

- How do we solve it.
  - Custom graph engine app.
  - Cache entire third degree for logged in members
  - Graph Algorithms
  - Minimize memory footprint of cache
    - Bit vector optimizations
    - compressions: p4Delta..
Linkedin Search

- Linkedin Search
  - 50 M documents
  - Business need to filter/rank with third degree cache
  - Response time: 30-50 ms.

- How do we solve it.
  - Distributed search engine.
  - Documents partitioned
    - 0-5M, 5-10M, ..
  - Each partition fetch network in range
  - Filter/Rank and return results.
Data Analytics

- Data is LinkedIn primary asset
- LinkedIn Data has immense value
  - Career trends
  - Bubble bursts
  - Hot keywords
  - Find experts/Influencers
- Big Data VS Better Algorithm
  - Big fight in machine learning community
  - Personal opinion: big data rocks if you know your data

Image1 source: http://otec.uoregon.edu/images/data-to-wisdom.gif
Image2 source: http://www.nature.com/ki/journal/v62/n5/images/4493262f1b.gif
How often do people change jobs?

Average Number of Early-Career Positions (first 10 yrs after graduating w/ a BS/BA)

- New York Area
- SF Bay Area
- Nationally

Graduation Year:
- 1971
- 1973
- 1975
- 1977
- 1979
- 1981
- 1983
- 1985
- 1987
- 1989
- 1991
- 1993
- 1995
- 1997

Y-axis:
- 1.5
- 1.7
- 1.9
- 2.1
- 2.3
- 2.5
- 2.7
- 2.9
- 3.1
- 3.3
- 3.5
People You May Know

- Shows People you **should** be connected to?
- For 50 M member
  - Potential $50^2 = 2500,000$ Billion
- Can be Narrowed down
  - Triangle closing
  - School/Company overlap
  - .. Other factors
  - secret sauce 😊
Scalable Architectures

• What is a scalable architecture?
  • Ability to do more if desired
  • With minimum effort
  • w/o software rearchitecture
• How to scale?
  • Scale Vertically (easy but costly)
  • Scale Horizontally (Harder but very scalable if done right)
• Divide & Conquer is the only way to scale horizontally
  • Sharding/Partitioning
Scalable Architectures

Checkpoint

Restore

Map/Reduce

Map
Reduce
Map
Reduce
Map
Reduce
Map
Reduce
Respect the elephant
“For the last several years, every company involved in building large web-scale systems has faced some of the same fundamental challenges. While nearly everyone agrees that the "divide-and-conquer using lots of cheap hardware" approach to breaking down large problems is the only way to scale, doing so is not easy “

Hadoop was born to fulfill this very need
What is Hadoop?

- An Open-source Java based implementation of distributed map-reduce paradigm
  - Apache project with very active community
    - heavy support from Yahoo
  - Distributed Storage + Distributed Processing + Job Management
  - Runs on Commodity Hardware (Heterogeneous cluster)
  - Scalable, Reliable, Fault tolerant and very easy to learn.
  - Yahoo currently running clusters with 10,000 nodes that processes 10TB of compressed data for their production search relevance processing
  - Written by Lucene author Doug Cutting (and others)
  - Plethora of valuable sub-projects
    - Pig, Hive, HBase, Mahout, Katta
What is Hadoop used for?

- **Search Relevance**: Yahoo, Amazon, Powerset, Zevents
- **Log Processing**: Facebook, Yahoo, Joost, Last.fm
- **Recommendation System**: Facebook
- **Data Warehouse**: Facebook, AOL
- **Video and Image Analysis**: New york times, eyealike
- **Prediction models**: Quantcast, Goldman Sachs
- **Genome Sequencing**: University of Maryland.

- **Academia**: University of Washington, Cornell, Stanford, Carnegie Mellon,
Why Use Hadoop

• Distributed/parallel programs are very very hard to write
• Tip of iceberg is the actual program/business logic
  • Map/Reduce code
• The hidden iceberg is
  • Parallelization (Divide & Conquer)
  • Data Management (TB to PB)
  • Fault Tolerance
  • Scalability
  • Data local optimizations
• Monitoring
• Job isolation
Hadoop core components

- HDFS distributed file system
  - User space distributed data management
    - Not a real file system (no POSIX, not in the kernel)
  - Replication
  - Rebalancing
  - Very high aggregate throughput
  - Files are immutable
- MapReduce layer
  - Very simple but powerful programming model
  - Move computation to the data
  - Cope with hardware failures
  - Non-goal: low latency
Map Reduce

- Google paper started it all:
  - “MapReduce: Simplified Data Processing on Large Clusters”, OSDI 2004
- **Map** = per “record” computation (extract/transform phase)
  - Each map task gets a block or piece of input data
- **Shuffle** = copy/collect data together (hidden from users)
  - Collect all values for same key together.
  - Can provide custom sort implementations.
- **Reduce** = final computation (aggregation/summarize phase)
  - All key, value pairs for one key goes to one reducer.
  - Reduce is optional
Map Reduce Example: WordCount

The overall MapReduce word count process

Input

Splitting

Mapping

Shuffling

Reducing

Final result

Deer Bear River

Car Car River

Deer Car Bear

Deer, 1
Bear, 1
River, 1

Car, 1
Bear, 1
River, 1

Deer, 1
Car, 1
Bear, 1

Bear, 2

Bear, 1

Car, 1

Car, 3

Car, 1

Car, 1

Deer, 2

Deer, 1

Deer, 1

River, 1

River, 2

River, 1

Map Reduce Implementation

• Jobs are divided into small pieces (Tasks)
  • Improves load balancing
  • Faster recovery from failed nodes

• Master
  • What to run?
  • Where to run?

• Slaves
  • One tasktracker per machine
  • Can configure how many mappers / reducers per tasktracker

• Manage task failures by re-starting on different node.
  • Speculative execution if a task is slow or have failed earlier

• Maps and reduces are single threaded Individual JVMs
  • Kill -9
  • Resource Isolation
  • Complete cleanup
Map Reduce

• **Job tracker**
  - splits input and assigns to various map tasks
  - Schedules and monitors map tasks (heartbeat)
  - On completion, schedules reduce tasks

• **Task tracker**
  - Execute map tasks – call mapper for every input record
  - Execute reduce tasks – call reducer for every intermediate key, list of values pair
  - Handle partitioning of map outputs
  - Handle sorting and grouping of reducer input
- **Namenode (master)**
  - Maps file Name to set of blocks
  - Maps blocks to list of datanodes where it resides.
  - Replication engine for blocks
  - Checksum based data coherency checker.
  - Issues
    - Metadata in memory (single point of failure)
    - Secondary name node checkpoints metadata.

- **Datanodes (slaves)**
  - Stores blocks on local file system.
  - Stores checksum of blocks
  - Read/write data to clients directly.
  - Periodically send a report of all existing blocks to Namenode.
HDFS Architecture

1. Client sends a filename to the NameNode.
2. NameNode returns a block id and list of DataNodes.
3. Client reads data from the DataNodes.

NameNode: Maps a file to a file-id and list of DataNodes.
DataNode: Maps a block-id to a physical location on disk.
Secondary NameNode: Periodic merge of Transaction log.
Hadoop @ Linkedin

- People You May Know
  - 40 individual jobs:
    - Graph analysis
    - School & Company overlap
    - Collaborative filtering
    - Misc. other factors
    - Regression model combines all values
- Content scoring
- Collaborative Filtering
- Sessionization and cause analysis
- Phrase analysis
- Other derived data: company data, matching fields, seniority, etc.
- Search relevance
  - Peoplerank
  - Link keywords to results based on user behavior
  - User bucketing
Data Visualization

- Can be very very useful.
  - Saves ton of time
  - Very insightful
  - Pictures = 1000+ words

- Challenges
  - Amount of data is huge
  - Need to write visualization for each problem.
  - Time spent per visualization is huge
  - Hard to make it distributed/scalable
Data Visualization
Questions
4. [http://www.youtube.com/watch?v=yjPBkvYh-ss](http://www.youtube.com/watch?v=yjPBkvYh-ss)
5. [http://www.youtube.com/watch?v=-vD6PUDf3Js](http://www.youtube.com/watch?v=-vD6PUDf3Js)
7. Linkedin.com
8. DJ Patil Talk at Scale Unlimited 2009