Grid Computing at Visvo using Hadoop

Dennis Kubes
Founder & CTO
Visvo, Inc.

Computing Trends

- **Mainframes – 1950s to 1960s**
  a. Automated processing shared single resource
  b. Transferring manual processes
- **Workstations – 1970s to 1980s**
  a. Each machine uses its own resources
  b. Came about because contention of resources and specialized problems
- **Personal Computers – 1980s to 1990s**
  a. Proliferation of hardware and cheap power microprocessors
  b. Personal computer client applications
  c. Applications have become more powerful over time in step with the capabilities of the hardware
- **Client Servers – 1990s to 2000s**
  a. Applications networked
  b. Problems are broken up into different functions (databases, web, etc.)
  c. Eventually became the internet paradigm (massive client-server)
- **Grid Computing – 2000s and beyond**
  a. Problems are too big for client/server
  b. Use many servers acting as one, break problems up into many parts that are run in parallel. Terabytes and greater.
  c. Applications that require extensive computing and storage resources. Grand challenge problems. Video and animation rendering. Search engines

Fetching Speeds

- 1 machine @ 10 pps = 1,000,000,000 / 10pps /60 seconds / 60 minutes / 24 hours / 365 days = 3.17 years

- 50 machines @ 10 pps = 1,000,000,000 / 500 pps / 60 seconds / 60 minutes / 24 hours = 23.1 days

- 500 machines @ 10 pps = 1,000,000,000 / 5000 pps / 60 seconds / 60 minutes / 24 hours = 2.3 days

Updating a Terabyte Database (10 billion rows)

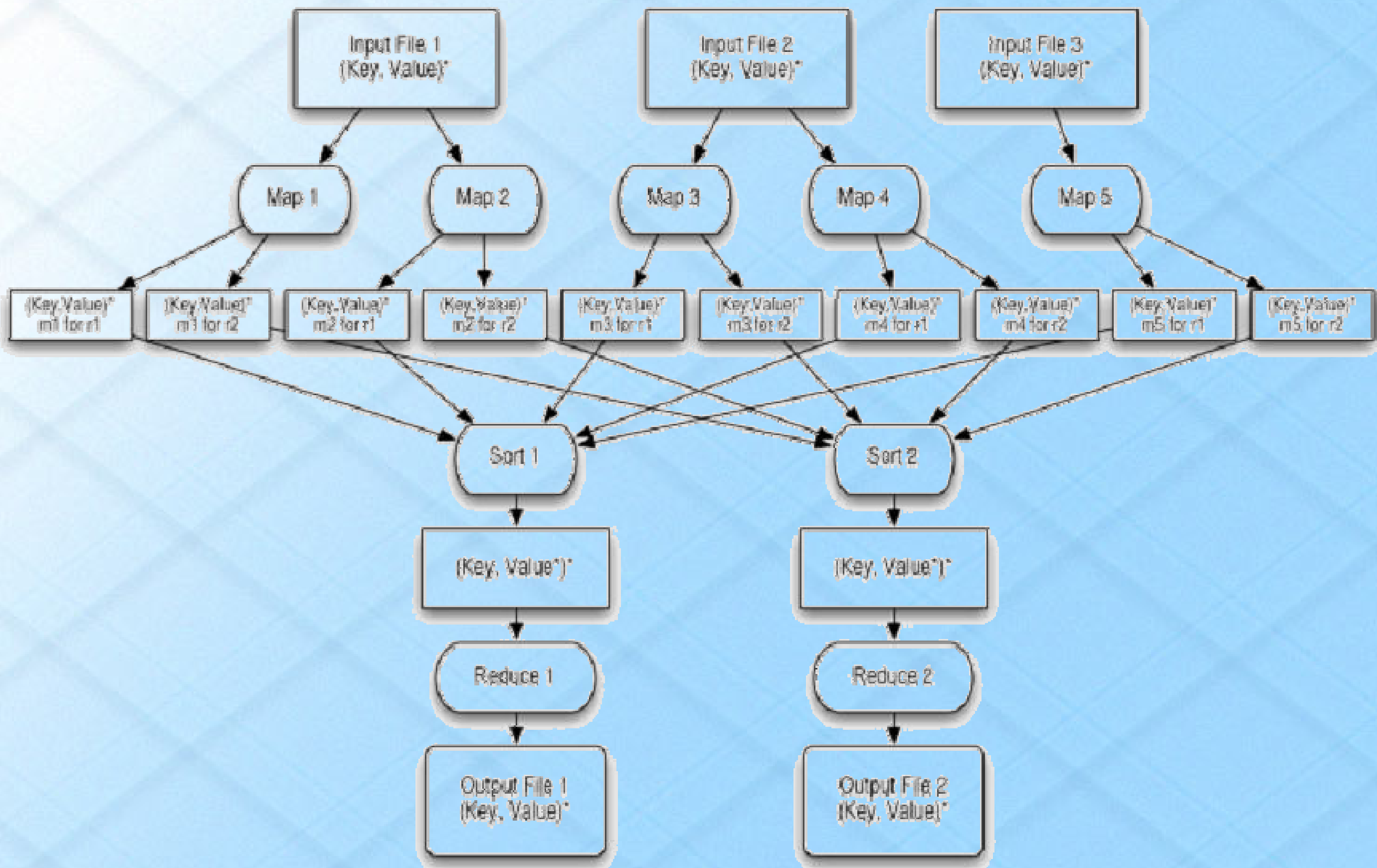
- Updating 1% of entries (100 million rows)

    a. 1000 days with random B-tree updates (regular updates)
    b. 100 days with batches B-tree updates (batches of updates no contention)
    c. 1 day (Grid based system)

The software grid – Hadoop

- Distributed File System
  a. Sizes of data
  b. Redundancy of data
  c. Self healing
- MapReduce
  a. Problem broken into multiple parts (divide and conquer)
  b. Each part is run on separate machines in parallel
  c. Results are aggregated together to produce output

The software grid – MapReduce



The hardware grid

- Blade Servers
- Commodity machines, processors, disks
- Reliability through software – expect machines to fail, software must handle gracefully
- Multi-core machines give greater computational ability
- Disk IO is the bottleneck
- Greater computational power (i.e. less time) through adding more commodity machines. Theoretically gives unlimited computing power and disk space.
- Amazon EC2/S3
  - Commission and decommission machines as you need them
  - Your own operating system and hardware
  - Only use resources when you need them (no dedicated hardware)

Grid Computing at Visvo

- 50 machines (Intel Core2Duo, 1U commodity, 4G RAM, 1x750G hard drive) = ~35T of space and 100 processors
- Fetching
  a. 500 crawlers -> 50 machines -> 10 Million pages/day
  b. Fetch 100 Million pages / 10 days
- Indexing
  a. Index 100 Million pages
- Categorization
  a. Categorize 100 Million pages into 150 categories

How to get started with Hadoop

- Pick a big problem that fits
  a) Something taking large computational or space resources
  b) Can be broken up into multiple similar pieces
  c) Image or file storage (file servers)
  d) Parsing large log files

- Use commodity hardware
- Or use a virtual hardware grid (Amazon EC2/S3)
- Different viewpoint for problem solving
  a) Batch jobs
  b) The level beyond databases and client server solutions

Resources

- Hadoop - http://lucene.apache.org/hadoop/
  a) http://wiki.apache.org/lucene-hadoop/
- Nutch - http://lucene.apache.org/nutch

Organizations using Hadoop

- Facebook
- Visvo
- Yahoo
- Last.fm
- Powerset

Credits

- Thanks to Owen O'Malley:
  - Some of the information for this presentation was taken from Owen O'Malley's ApacheCon 07 Hadoop presentation .
  - http://wiki.apache.org/lucene-hadoop-data/attachments/HadoopPresentations/attachments/HadoopApacheConEu07.pdf
- Thanks to the Hadoop Team