Pig as a Solution for Accessing Peta-scale Astronomical Datasets

Ivan Zolotukhin

Paris Observatory / Sternberg Astronomical Institute

ADASS XX, Boston
November 8, 2010
About me

- Researcher at Paris Observatory
- Concerned with data intensive science
- Large-scale IT projects in the past
- Keep close contacts with industry leaders
Data intensive astronomy

- Exponential data growth
- Public access is crucial for project’s success
- Challenge to provide it
- Hence, clear demand for proper technology
Requirements

- Linear scalability on astronomical problems
- Easy to learn query language
- Low cost (hardware and software)
What’s already there

- Parallel databases
Parallel databases are prohibitively expensive
Parallel databases are prohibitively expensive
Open source RDBMSes
- Parallel databases are prohibitively expensive
- Open source RDBMSes need huge lifting
Parallel databases are prohibitively expensive
Open source RDBMSes need huge lifting
SDSS CasJobs backend
Parallel databases are prohibitively expensive
Open source RDBMSes need huge lifting
SDSS CasJobs backend: fine, but Microsoft
What’s already there

- Parallel databases are prohibitively expensive
- Open source RDBMSes need huge lifting
- SDSS CasJobs backend: fine, but Microsoft
- ScienceDB
What’s already there

- Parallel databases are prohibitively expensive
- Open source RDBMSes need huge lifting
- SDSS CasJobs backend: fine, but Microsoft
- ScienceDB: not really there yet
There’s something though

Hadoop
Hadoop

- **HDFS**, distributed filesystem
- **MapReduce**, framework for parallel computation
- Open source, Apache top level project
- Scalable: Yahoo! has 4000+ node cluster
MapReduce

- Google’s distributed computing framework
- Generic key-value based interface
- Split task into small pieces executed in parallel
- Aggregate everything in the end
- Most of data access problems can be decomposed into MapReduce jobs
Yuji Shirasaki (NAOJ) and pals
Cross-match of 1B records from largest catalogs
Hadoop cluster with 10 nodes
Took only 3.7 hours!
JVO experiment

Seems to scale linearly till number of hard disks

- Experiment
- $t_1/n$ (expectation for zero time lag, no competition among the threads)

Number of total CPU = 80 Cores
Number of total HDD = 40 (10xRAID5)
Hadoop is great, but...

- Analysis is done in Java
- Joins, filters lead to custom code
- Lengthy and error prone
- Query requires code compilation
- High level language
- Scan-centric approach
- Transformations on a sets of records
- Process data step by step
- Is probably easier than SQL
- Supports UDFs
- Developed at Yahoo!, now open source
Why Pig?

Because I bet you can read the following Pig script:

```pig
users = load 'users.csv' as (username: chararray, age: int);
users_1825 = filter users by age >= 18 and age <= 25;

pages = load 'pages.csv' as (username: chararray, url: chararray);

joined = join users_1825 by username, pages by username;
grouped = group joined by url;
summed = foreach grouped generate group as url, COUNT(joined) AS views;
sorted = order summed by views desc;
top_5 = limit sorted 5;

store top_5 into 'top_5_sites.csv';
```
Why Pig?

The same in Hadoop MapReduce
Why Pig?

The same in Hadoop MapReduce
Why Pig?

- Democratizes large-scale data analysis
- 5% of the code
- 5% of the time
- Within 50% of the execution time
People need CasJobs, but for everybody
Seems that Pig/Hadoop is a good candidate
Not only PB, also GB!
Anyone has hardware for experiments?
Thanks for listening

Credits

- Kevin Weil, Twitter
- VO-Paris guys: Joel Marchand, Pierre Le Sidaner