Analyzing Massive Astrophysical Datasets
Can Pig/Hadoop or a relational DBMS help?

Sergey Negrashov

University of Hawaii at Manoa
Information & Computer Sciences

Jan 16th 2013

THE N-BODY PROBLEM IN CLASSICAL MECHANICS IS STILL UNSOLVED.

WE DON’T ACTUALLY KNOW IF OUR SOLAR SYSTEM IS STABLE OR NOT.

STILL, WE WOULDN’T WAKE UP ONE MORNING TO FIND A PLANET MISSING.

BUT THAT IS EXACTLY WHAT HAPPENED TO PLUTO!
Outline

Introduction
Astrophysics Simulation
Hadoop
Showdown
N-body Problem

N-body problem is the problem of predicting a future state of a gravitational system from initial conditions.

N-Body Problem does not have an analytic solution for $N > 3$:

- Computer science: NP-complete
- Math: Galois Theorem
- Astrophysics: Numerical Solution
N-body Problem Cont.

N-body simulation parameters:
- $N$: number of bodies
- $\Delta t$: simulation timestep
- $T_{\text{final}}$: simulation length
- $T_{\text{checkpoint}}$: checkpoint frequency

N-body simulation workflow:

1. $t \leftarrow 0$
2. while $t < T_{\text{final}}$ do
3. for particle $p$ in $N$ do
4. Compute all forces on $p$
5. Compute position at $t + \Delta t$
6. end for
7. $t = t + \Delta t$
8. if $(t - t_{\text{last}}) > T_{\text{checkpoint}}$ then
9. Create checkpoint for $t$
10. $t_{\text{last}} \leftarrow t$
11. end if
12. end while
## Astrophysics Simulation Scale

<table>
<thead>
<tr>
<th>Name</th>
<th># of Particles</th>
<th>Snapshot Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbtest128g</td>
<td>$4.2 \times 10^6$</td>
<td>169 MB</td>
</tr>
<tr>
<td>cosmo50</td>
<td>$3.36 \times 10^7$</td>
<td>1.4GB</td>
</tr>
<tr>
<td>como25</td>
<td>$9.168 \times 10^8$</td>
<td>3.6GB</td>
</tr>
</tbody>
</table>

Visible universe contains $\approx 10^{22}$ stars or $\approx 10^{80}$ atoms.
**Simulation Analysis**

**Strategy:**
Analyze checkpoint files.

**Tools:**
- IDL
- Python/Perl
- C++ libraries

**Trend:**
- Constraints on CPU IO RAM
- Distributed memory model?
Could we use a Parallel DBMS or MapReduce?

If the problem can be partitioned correctly MapReduce takes care of the parallelism for you.

VS

Parallel DBMS tackle similar problems in data-mining. These systems could be adapted to the Scientific workload.
Hadoop:
- An open source MapReduce implementation
- API for Java, C++, python ...
- Pig Programming language
- HDFS distributed filesystem
Queries:

**Q1:** Return all particles whose property $X$ is above a given threshold at step $S_1$.

$$\pi(iOrder)\sigma(x > T)(S_1)$$

**Q2:** Return all particles of type $T$ within distance $R$ of point $P$.

$$\pi(iOrder)\sigma(p.x^2 + p.y^2 + p.z^2 > R^2)(S_1)$$

**Q3:** Return all particles of type $T$ within distance $R$ of point $P$ whose property $X$ is above a threshold computed at timestep $S_1$.

$$\pi(iOrder)\sigma(p.X > T) \& (p.x^2 + p.y^2 + p.z^2 > R^2)(S_1)$$

**Q4:** Return gas particles destroyed between step $S_1$ and $S_2$.

$$\pi(iOrder)\sigma\ p_1.iOrder \notin S_2 (S_1 \times S_2)$$

**Q5:** Return all particles whose property $X$ changes from $S_1$ to $S_2$.

$$\pi(iOrder)\sigma(p_1.iOrder = p_2.iOrder) \& (p_1.X \neq p_2.X)(S_1 \times S_2)$$
Single Node Performance:

**Specs:**
- IDL: 16CPU, 128 GB RAM
- DBMS/Hadoop: 8CPU, 16GB RAM.

**Notes:**
Hadoop and IDL had to read in the snapshot file every time.
DBMS had indexed data available to it.
Single Node Large Dataset:

Query Performance on 32GB dataset.
Multi-Node Performance:

- Query: Q1, Q2
- Query: Q3
- Query: Q4
- Query: Q5
Questions?