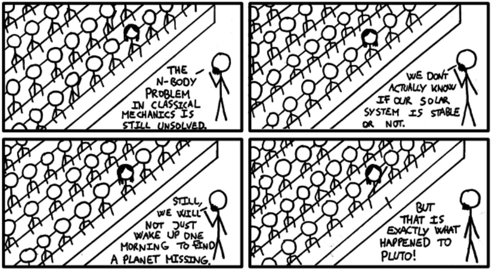
Analyzing Massive Astrophysical Datasets

Can Pig/Hadoop or a relational DBMS help?

Sergey Negrashov

University of Hawaii at Manoa
Information & Computer Sciences

Jan 16th 2013



Outline

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
- Δt: simulation timestep
- T_{final}: simulation length
- $T_{checkpoint}$: checkpoint frequency

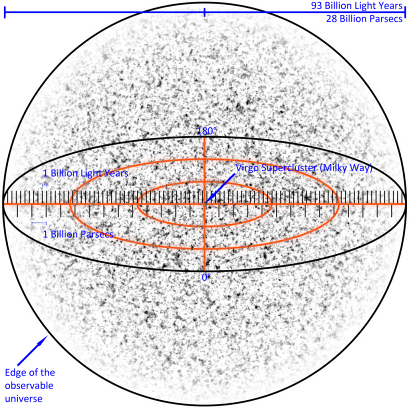
N-body simulation workflow:

$t \leftarrow 0$
while $t < T_{final}$ **do**
    for particle p in N **do**
        Compute all forces on p
        Compute position at $t + \Delta t$
    end for
    $t = t + \Delta t$
    if $(t - t_{last}) > T_{checkpoint}$ **then**
        Create checkpoint for t
        $t_{last} \leftarrow t$
    end if
end while

Astrophysics Simulation Scale

Name	# of Particles	Snapshot Size
dbtest128g	$4.2 * 10^6$	169 MB
cosmo50	$3.36 * 10^7$	1.4GB
como25	$9.168 * 10^8$	3.6GB

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



Simultion Analisys

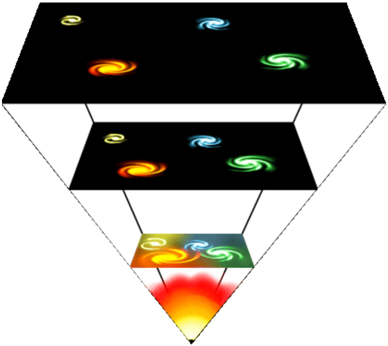
Strategy:
Analyze checkpoint files.

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

Hadoop:

- An open source MapReduce implementation
- API for Java, C++, python ...
- Pig Programing 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 timstep 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:

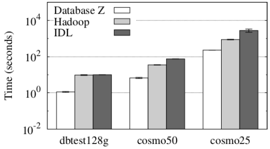
Specs:

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

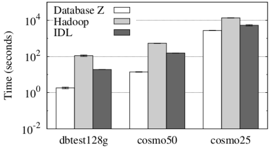
Notes:

Hadoop and IDL had to read in the snapshot file every time.

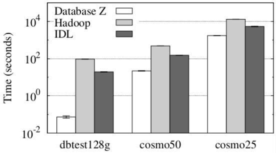
DBMS had indexed data available to it.



Query: Q1,Q2,Q3

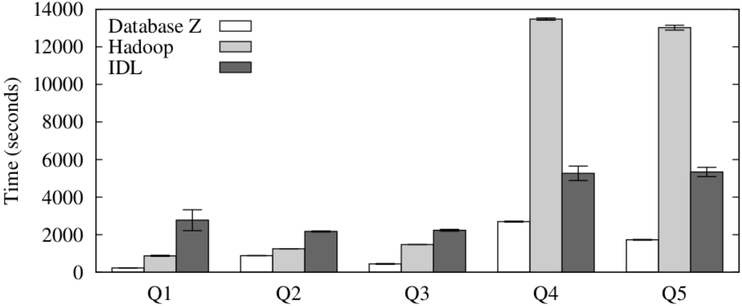


Query: Q4



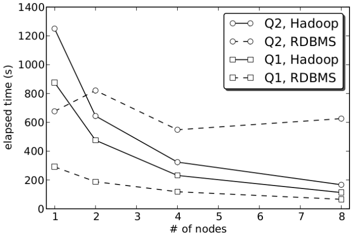
Query: Q5

Single Node Large Dataset:

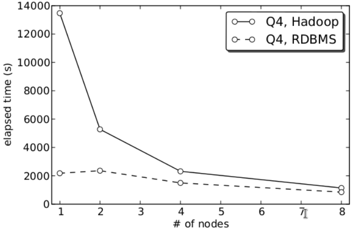


Query Performance on 32GB dataset.

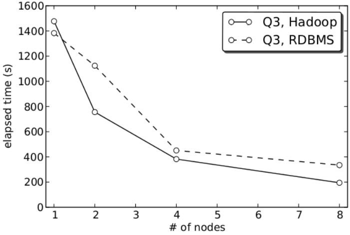
Multi-Node Performance:



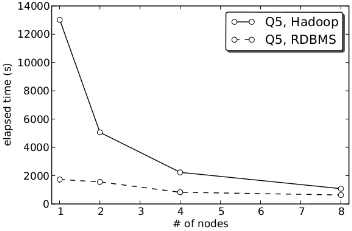
Query: Q1,Q2



Query: Q4



Query: Q3



Query: Q5

Questions?