Unified Relational GIS: Workloads, Schemas, Early Performance Results

Peter A. Dinda, Northwestern University

http://www.cs.northwestern.edu/~pdinda

Collaborator: Beth Plale, Indiana University

Claim

Applications need *common compositional* queries over information of *varying dynamicity*

Approach

Build down from an RDBMS world-view

Relational = relational data model and queries

Unified = tables and streams

Research Questions

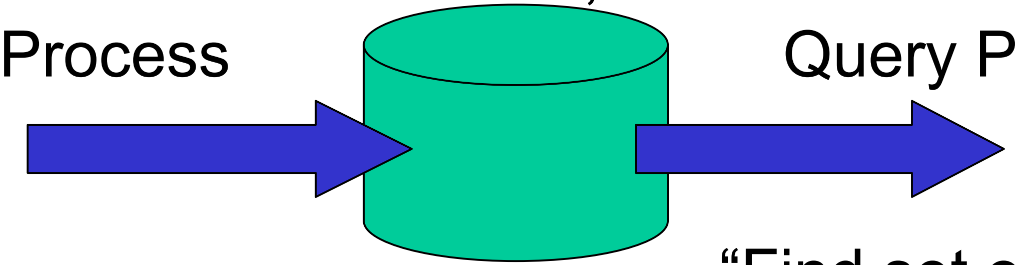
How "far down" must we go?

What extensions are needed?

Outline

- Workloads
    - Host Tiers
- Schema and development and implementation
    - RGIS1, RGIS2
- Initial performance results
    - Update rates (RGIS1, RGIS2)
    - Non-deterministic queries (RGIS1)
    - Deterministic queries (RGIS2)

Workload



Update Process

Database
Schema
RDBMS, etc

Query Process

“Host itchy has been upgraded to 2 GB”

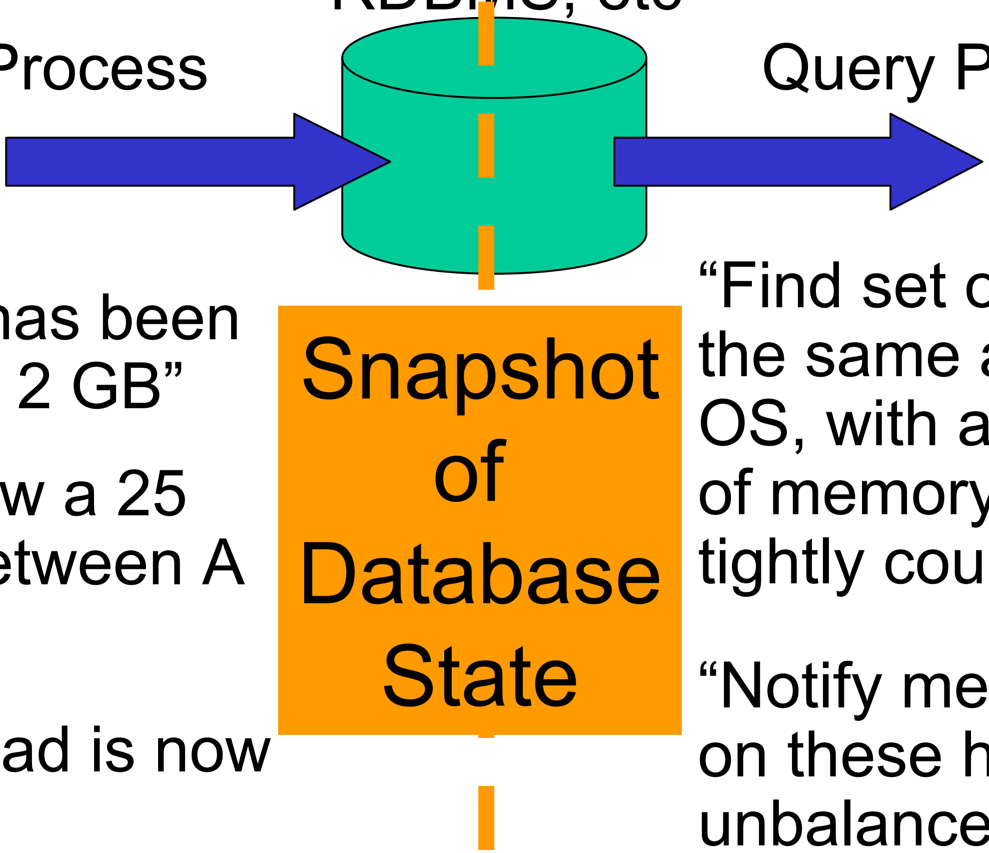
“There is now a 25 gbit/s link between A and B”

“Host A’s load is now 0.9”

“Find set of four hosts, all the same architecture and OS, with a total of 10 GB of memory, that are on a tightly coupled network”

“Notify me when the load on these hosts becomes unbalanced”

Current Workload Modeling



Database
Schema
RDBMS, etc

Update Process

Query Process

“Host itchy has been upgraded to 2 GB”

“There is now a 25 gbit/s link between A and B”

“Host A’s load is now 0.9”

Snapshot
of
Database
State

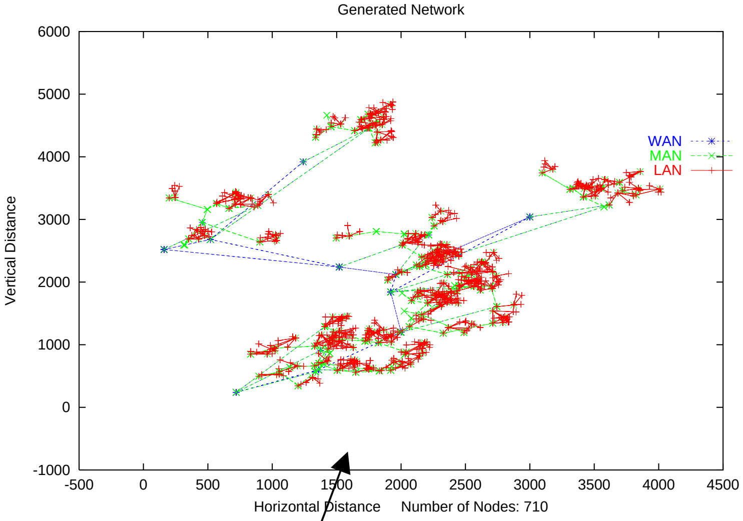
“Find set of four hosts, all the same architecture and OS, with a total of 10 GB of memory, that are on a tightly coupled network”

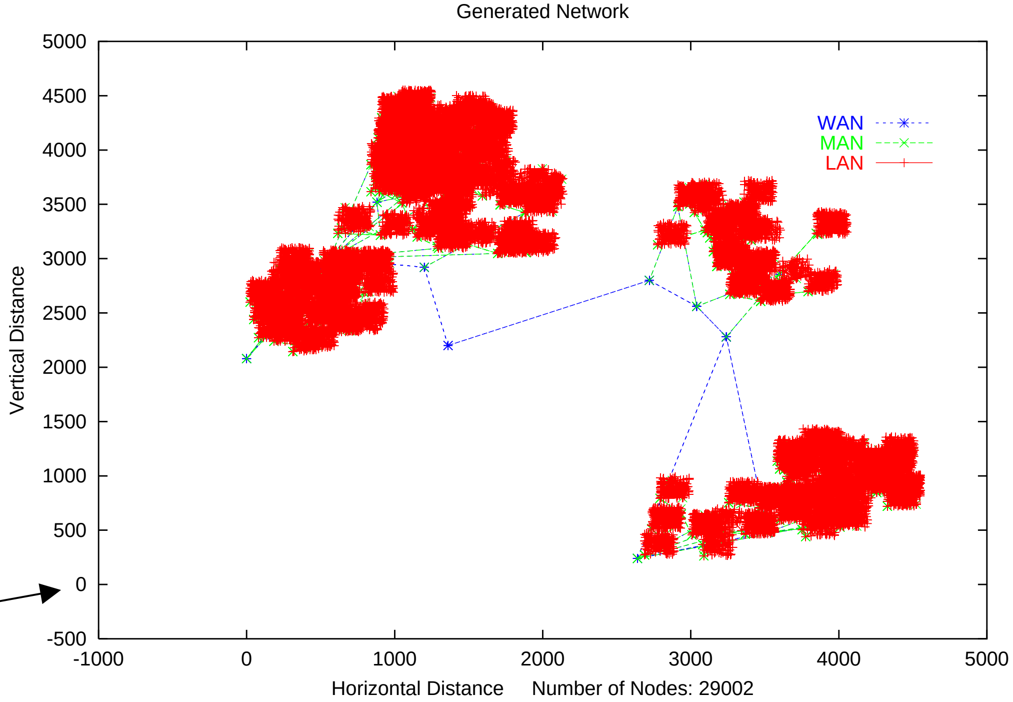
“Notify me when the load on these hosts becomes unbalanced”

Host Tiers (Student: Dong Lu)

- Tiers: Extant network topology generator
  - Randomly generated network graphs with constraints
- Extension: annotate graph with relevant network and host properties
  - “Grid Generator”
- Little is known about distribution or correlation of such properties.
  - Current Host Tiers assumes no correlations and uses relatively simple “intuitive” distributions of CPU, RAM, Disk, and network properties

Host Tiers Output





Randomly generated graphs annotated with randomly generated numeric and categorical data, all following constraints

Graphs and annotations are inserted into the database

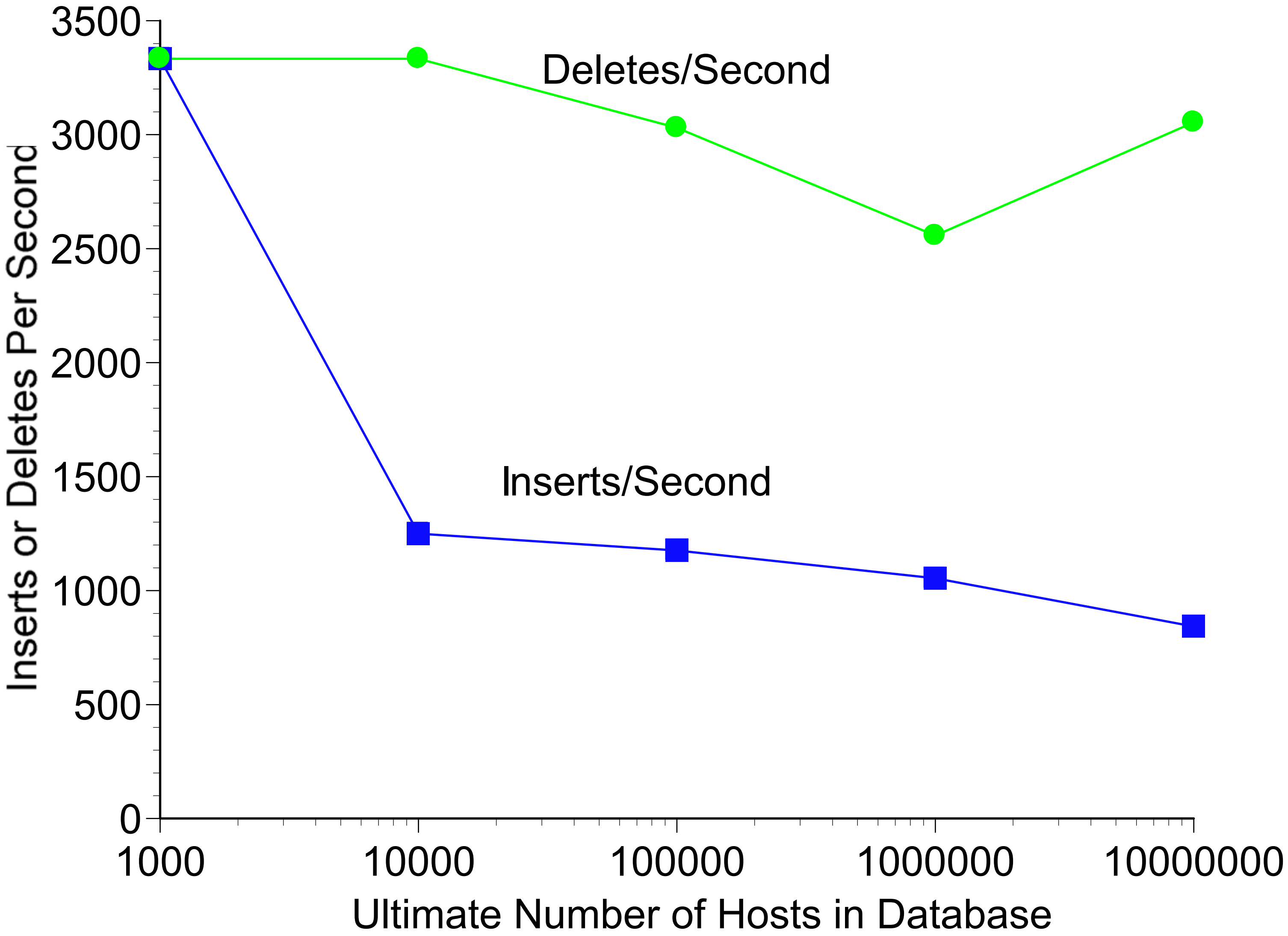
178 seconds for 1 million hosts, 3200 routers, 1.5 million links

RGIS1

- Physical resources: hosts, routers
- Software resources: executables
- Dynamic resources: connectivity of running distributed applications
- Benchmarks: performance tests

- Implemented on MySQL + Perl
- Available on web site

RGIS1 Insert/Delete Performance



PowerEdge 4400 (2x 1 GHz Xeon, 2 GB, 240 GB RAID)
RGIS1, MySQL, single inserts

Non-deterministic Time-bounded Queries

- Queries can be incredibly expensive
  - N-way joins
- Typically don’t need “all the answers”
  - Example: “Find 4 hosts which all have the same architecture and have a combined memory of 0.5 to 1 GB”
  - Only one such group is needed
- Typically have time and resource constraints

Run until the deadline, returning a non-deterministic subset of the full query results

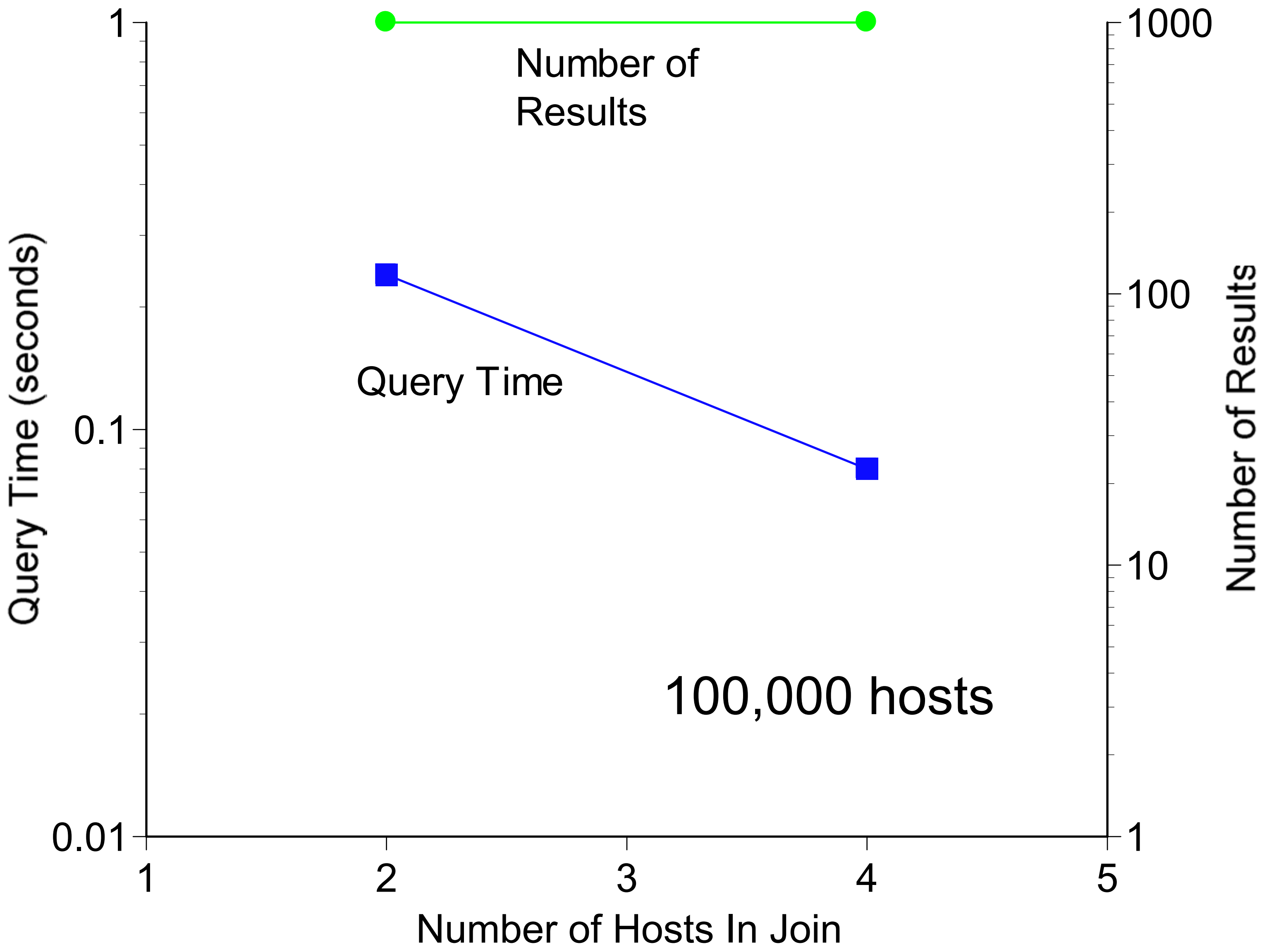
Example

```
select nondeterministically
   host1.name, host2.name, host3.name, host4.name,
   hd1.mem+hd2.mem+hd3.mem+hd4.mem as TotalMem,
from
   hosts as host1, hostdata as hd1,
   hosts as host2, hostdata as hd2,
   hosts as host3, hostdata as hd3,
   hosts as host4, hostdata as hd4
where
   host1.ip=hd1.ip and  host2.ip=hd2.ip and
      host3.ip=hd3.ip and host4.ip=hd4.ip and
   hd1.mem+hd2.mem+hd3.mem+hd4.mem>=512 and
   hd1.mem+hd2.mem+hd3.mem+hd4.mem<=1024 and
   host1.ip!=host2.ip and host1.ip!=host3.ip and
      host1.ip!=host4.ip and host2.ip!=host3.ip and
      host2.ip!=host4.ip and host3.ip!=host4.ip
order by
   TotalMem desc
limit
   1
inlessthan
   5 seconds
usingheuristic
   prefer_depth_first
```

Implementation of Non-deterministic, Time-bounded Queries

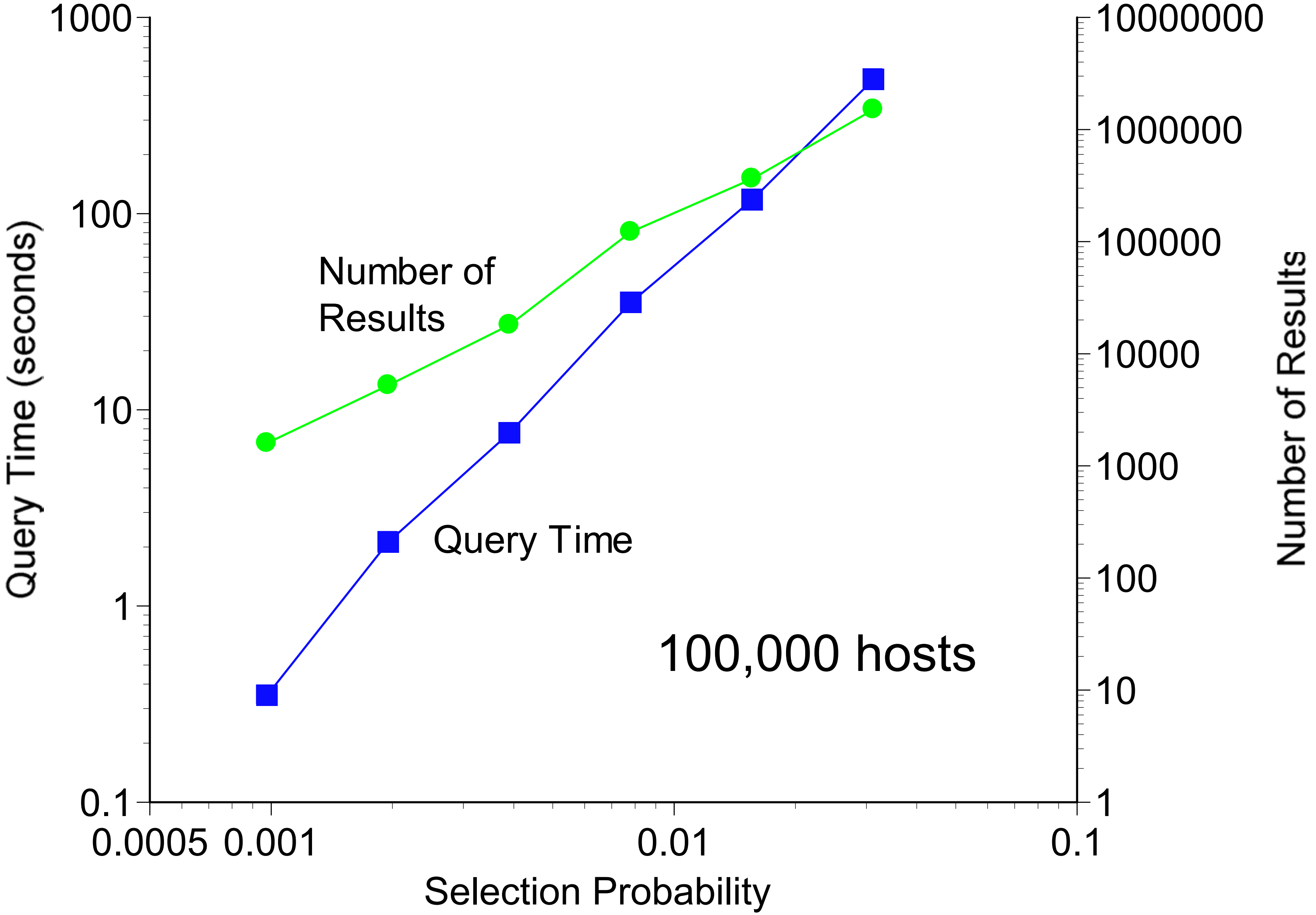
- Random number associated with each row in each table (or insert)
- Query is rewritten to incorporate a random ranges on the input tables
- Range lengths chosen to meet deadline
  - This is not trivial and we don’t have this translation yet
- Heuristics not yet incorporated
- Hopefully RDBMS-independent

RGIS1 Non-deterministic Query Performance



Find n hosts with a total memory of 1 GB of memory

RGIS1 Non-deterministic Query Performance



Find 2 hosts with a total memory of 1 GB of memory

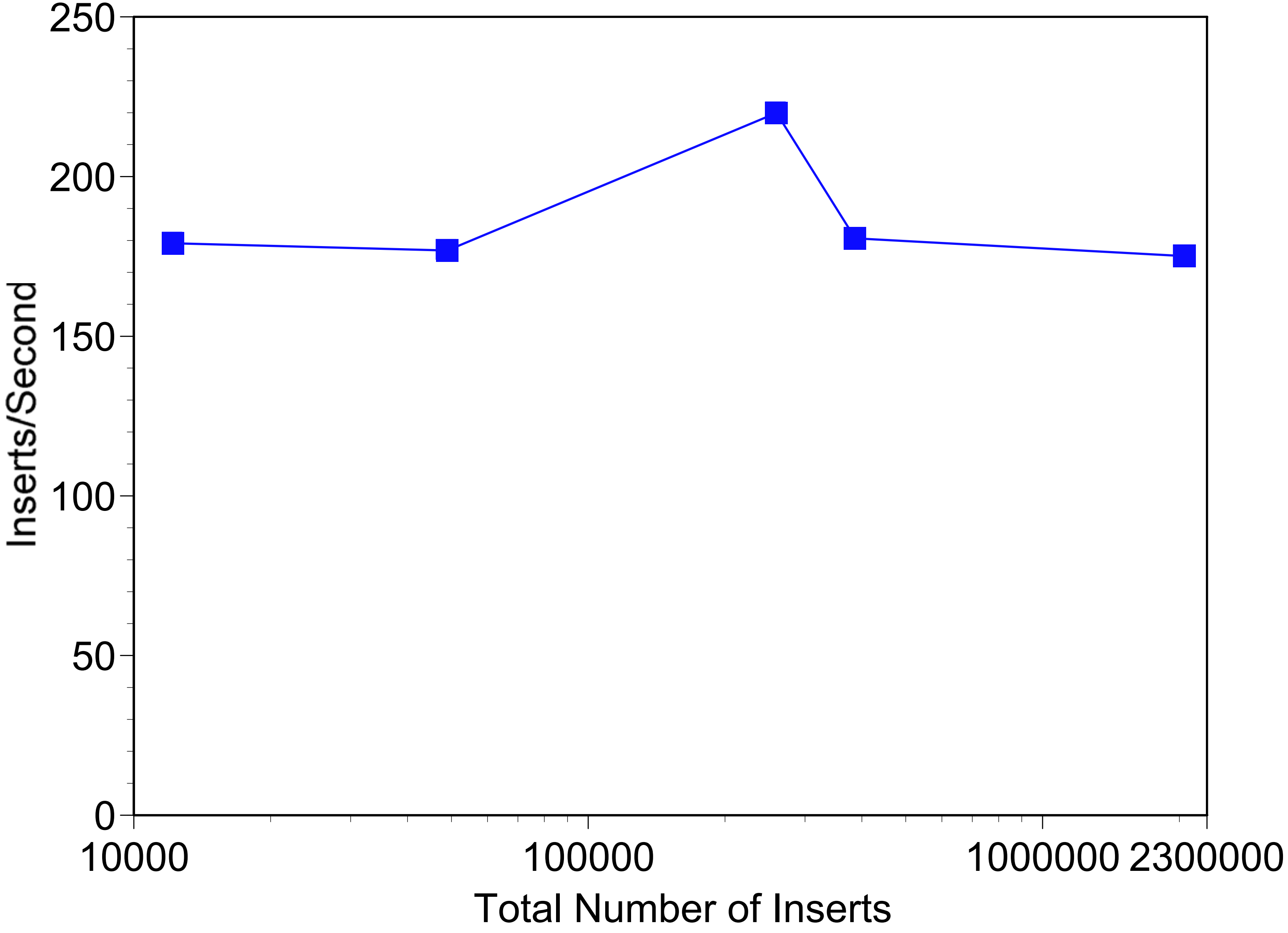
RGIS2

- Models network at layers 3, 2, and “1”
- Type information and separately managed type tables
- Strongly constrained data model
- Updates are now fully transactional and uniquely tagged
- Updates tagged with random numbers for non-deterministic queries

RGIS2

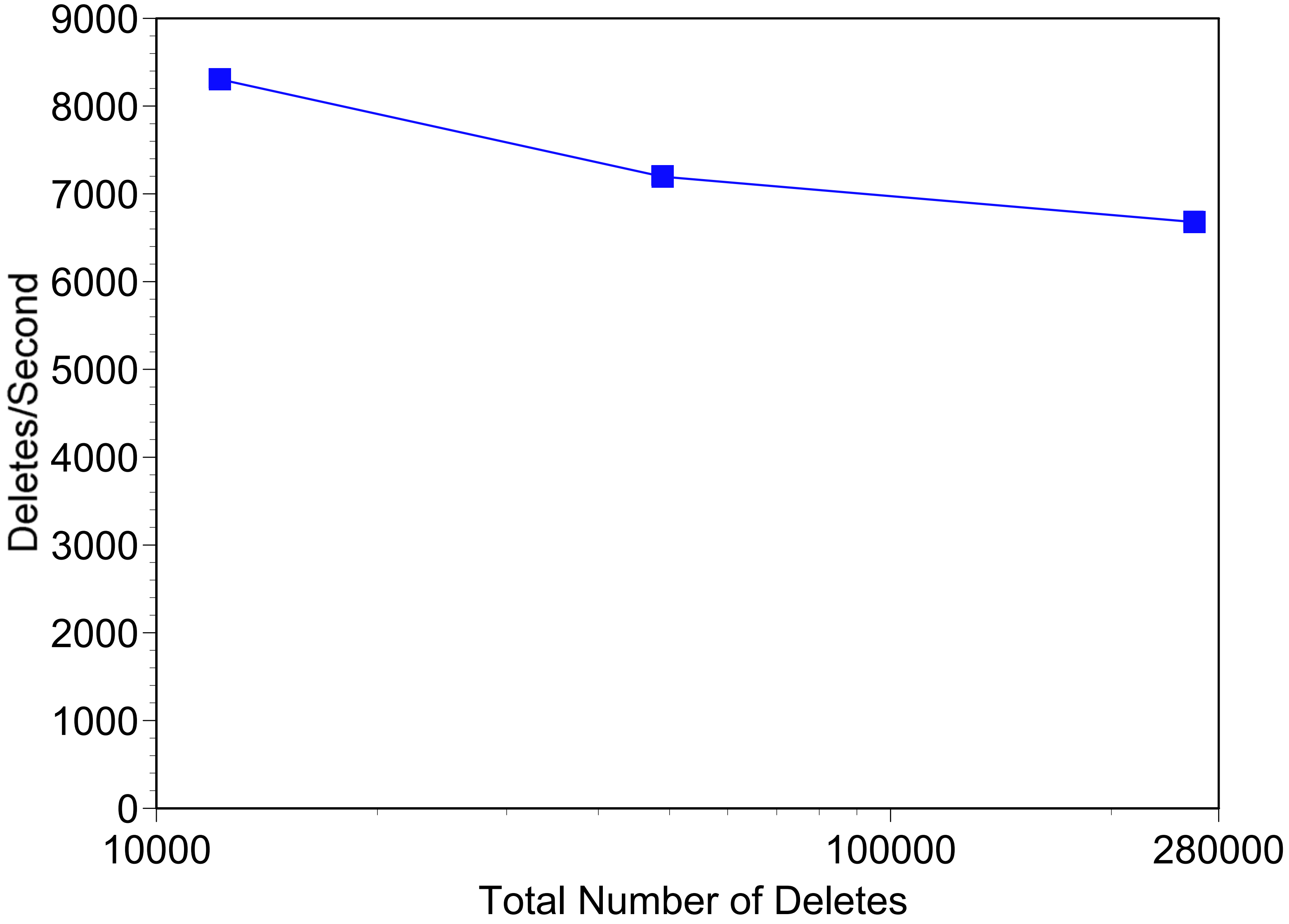
- Implemented on Oracle 9i
  - SQL, PL/SQL, Perl, C++
  - Use Oracle graph and procedural features
- Web interface / web services model in progress (OGSA?)

Very Preliminary RGIS 2 Insert Performance

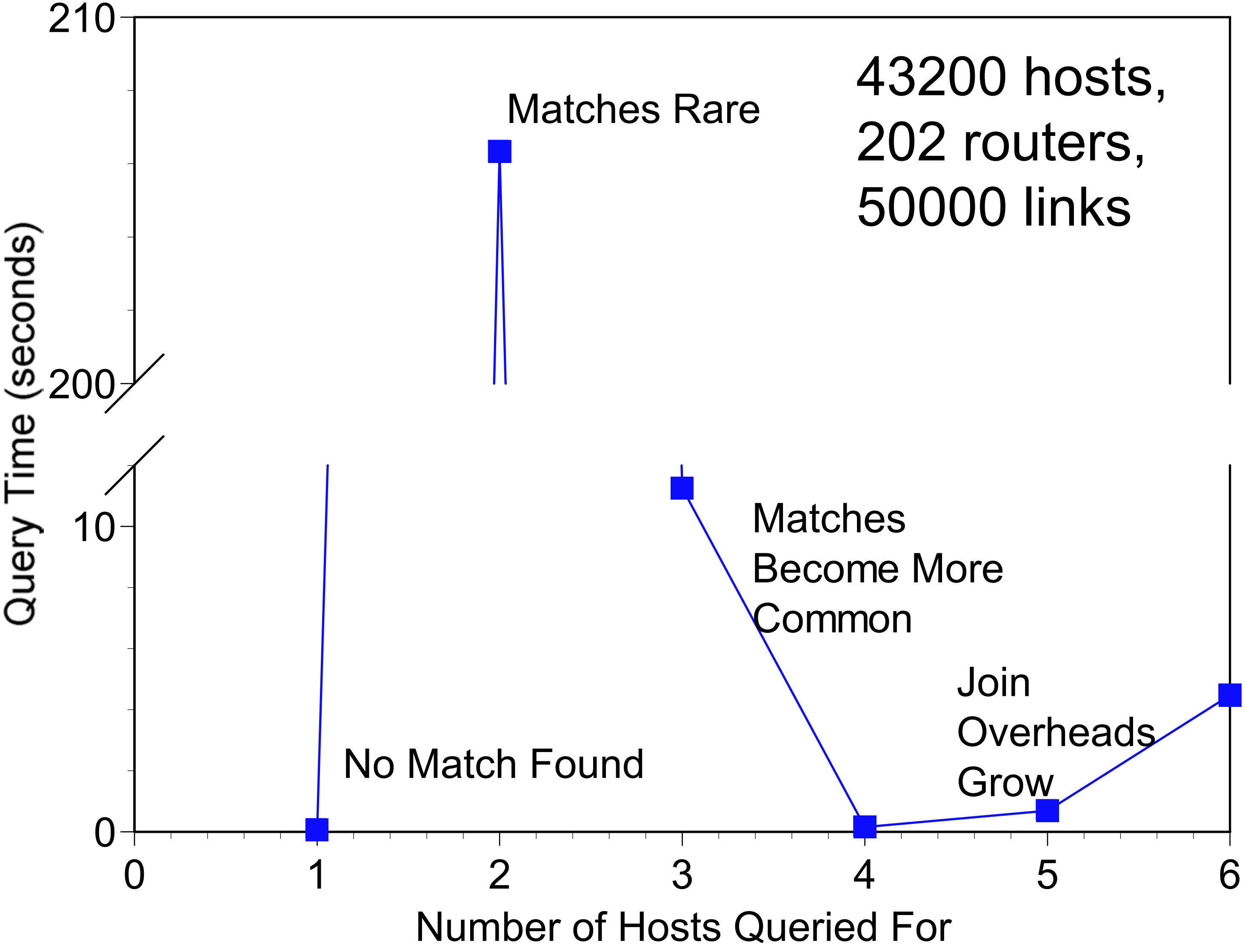


PowerEdge 4400 (2x 1 GHz Xeon, 2 GB, 240 GB RAID)
RGIS2, Oracle 9i Enterprise, Host Tiers, single inserts

Very Preliminary RGIS 2 Delete Performance



Very Preliminary RGIS2 Deterministic Query Performance



Find n hosts with a total memory of 3.2 GB, total speed of 4 GHz, all IA32, all running RH Linux, limit to 6 matches

Conclusions

- Workloads are critical, but the GIS community has very few
  - Potential for synthetic “grid generators” like Host Tiers
  - NEED MORE DATA
- Nascent Relational GIS implementation
  - On second generation now
  - Non-deterministic queries
  - Take performance results with grain of salt
    - Especially RGIS2: work in progress, limited indexing, etc