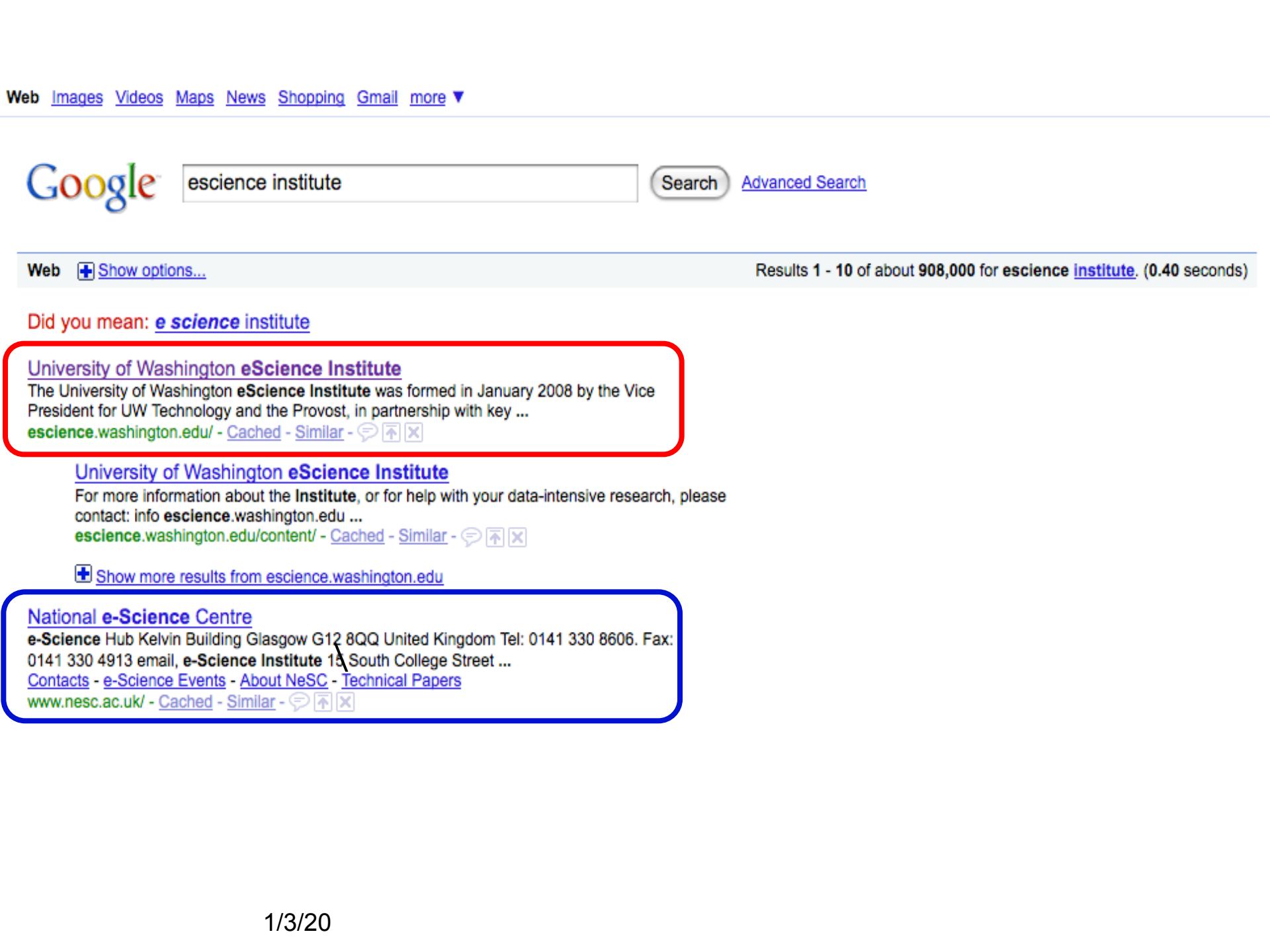
Data-Intensive Scalable Science: Beyond MapReduce

Bill Howe, UW

…plus a bunch of people


eScience Institute

http://escience.washington.edu


Web Images Videos Maps News Shopping Gmail more ▼
Google
escience institute
Search
Advanced Search
Web
Show options...
Results 1 - 10 of about 908,000 for escience institute. (0.40 seconds)
Did you mean: e science institute
University of Washington eScience Institute
The University of Washington eScience Institute was formed in January 2008 by the Vice President for UW Technology and the Provost, in partnership with key ...
escience.washington.edu/ - Cached - Similar -
University of Washington eScience Institute
For more information about the Institute, or for help with your data-intensive research, please contact: info escience.washington.edu ...
escience.washington.edu/content/ - Cached - Similar -
Show more results from escience.washington.edu
National e-Science Centre
e-Science Hub Kelvin Building Glasgow G12 8QQ United Kingdom Tel: 0141 330 8606. Fax: 0141 330 4913 email, e-Science Institute 15 South College Street ...
Contacts - e-Science Events - About NeSC - Technical Papers
www.nesc.ac.uk/ - Cached - Similar -

ALL RESULTS

Were you looking for: ***science** institute*

RELATED SEARCHES

- Science **Museum**
- **Franklin** Institute Science **Museum**
- **Search** Institute
- Science **Center**
- **Skin** Science Institute
- **Reproductive** Science Institute
- **Health** Sciences Institute
- **Brick Computer** Science Institute

SEARCH HISTORY

- escience institute
- escience institute washington

ALL RESULTS 1-10 of 87,700,000 results · Advanced

Institute of Noetic **Sciences**: Home Page
IONS is a nonprofit membership organization that conducts and sponsors leading-edge research into the potentials and powers of consciousness: including perceptions, beliefs ...
www.noetic.org · Cached page

Virginia **Institute** of Marine **Science** - Home
An **institution** that is uniquely prepared to educate the highly qualified researchers, resource managers, and educators needed for the future. Gloucester Point, Virginia.
www.vims.edu · Cached page

ICSI: International Computer **Science Institute** | Berkeley, California
International Computer **Science Institute** at the University of California, Berkeley. Research groups and technical reports.
http.icsi.berkeley.edu · Cached page

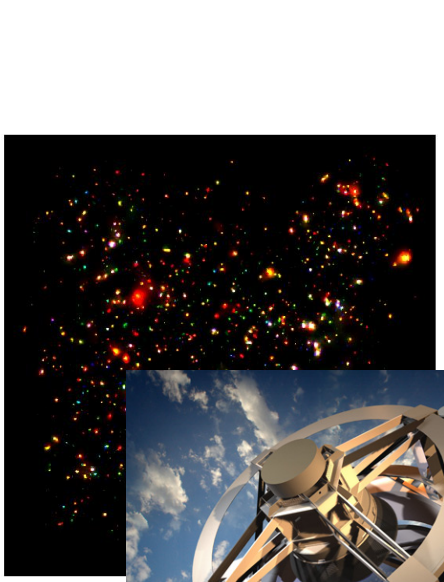
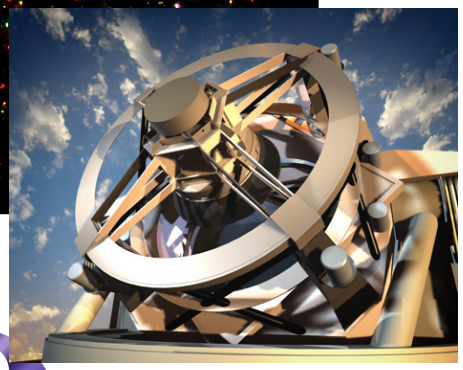
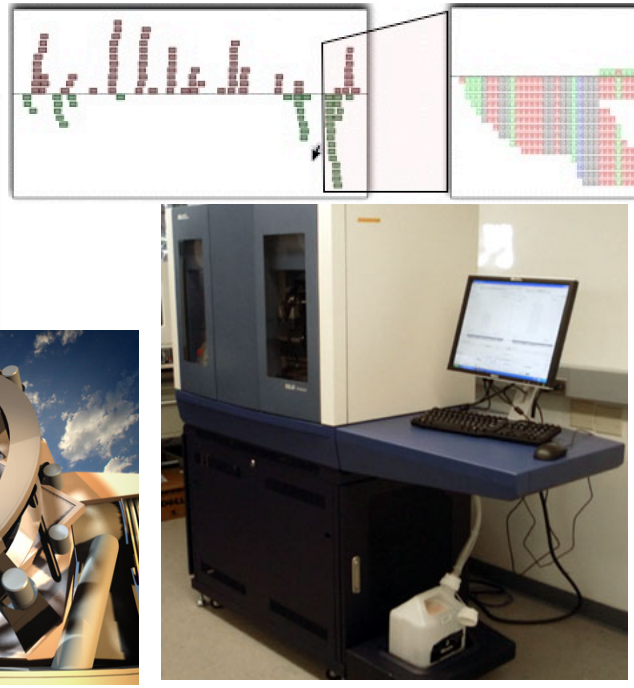
National **Institute** for Discovery **Science** (NIDS): **Science institute** ...
The National **Institute** for Discovery **Science** (NIDS) is a privately funded **science institute** engaged in research of UFOs, animal mutilations, and other related anomalous phenomena.
www.nidsci.org · Cached page

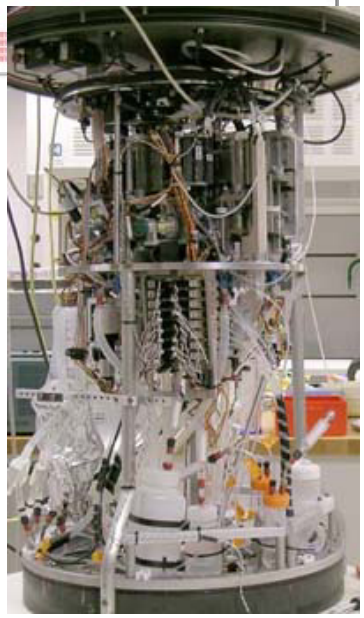
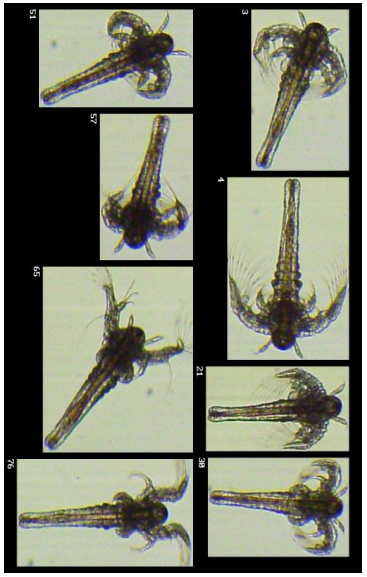
Science is reducing to a database problem

Old model: "Query the world" (Data acquisition coupled to a specific hypothesis)

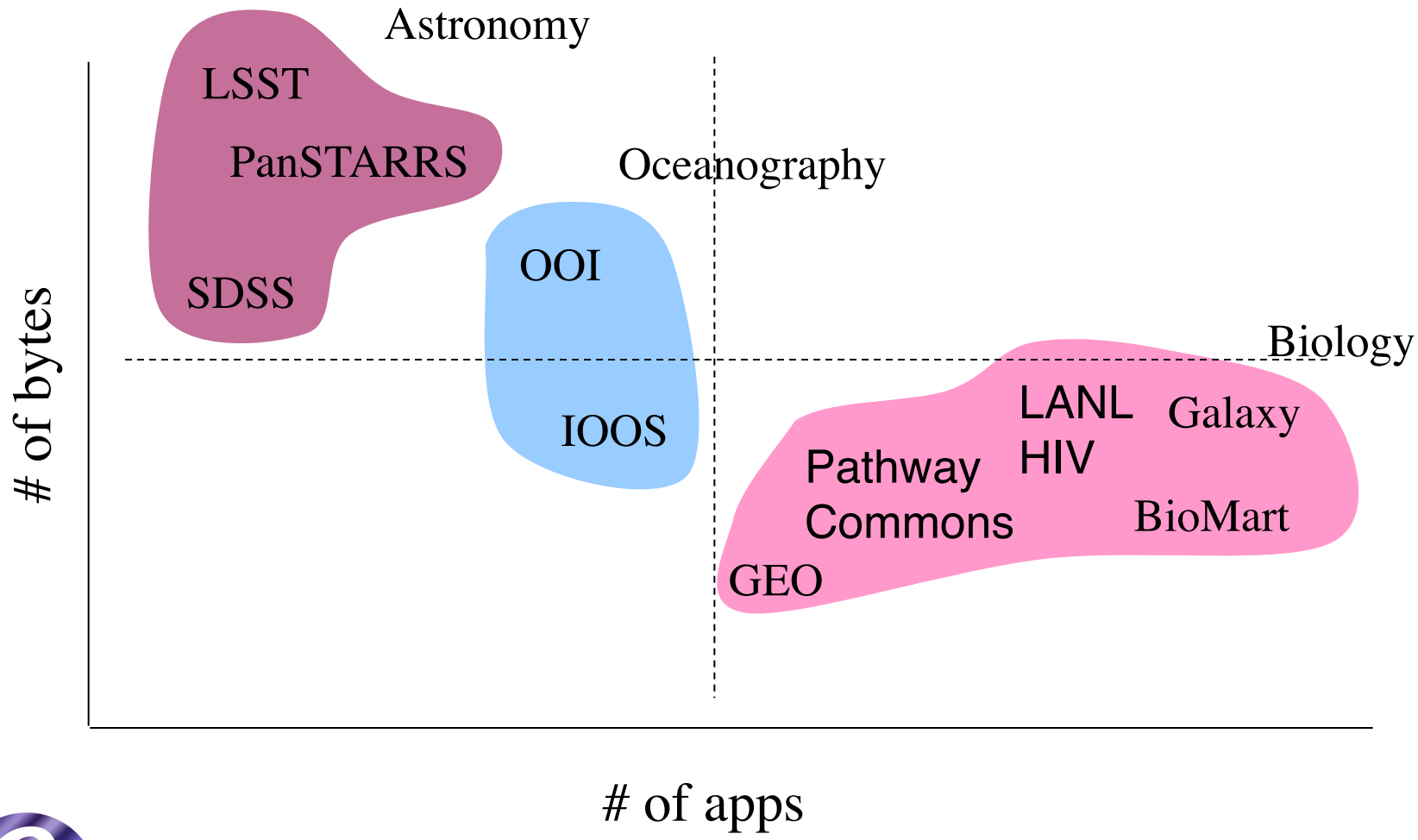
New model: "Download the world" (Data acquired en masse, in support of many hypotheses)

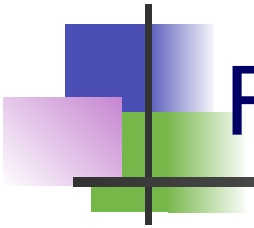
- Astronomy: High-resolution, high-frequency sky surveys (SDSS, LSST, PanSTARRS)
- Oceanography: high-resolution models, cheap sensors, satellites
- Biology: lab automation, high-throughput sequencing,

Two dimensions





Roadmap

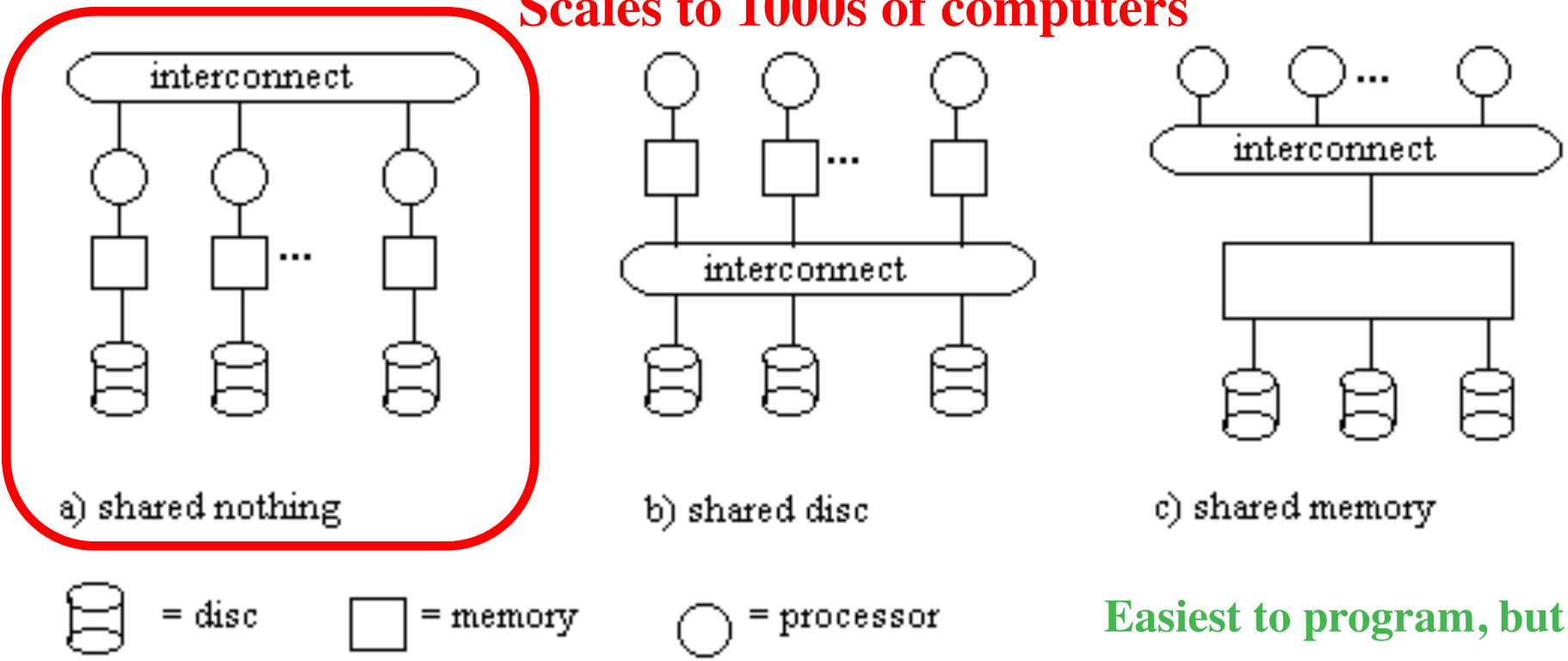
- Introduction
- Context: RDBMS, MapReduce, etc.
- New Extensions for Science
  - Spatial Clustering
  - Recursive MapReduce
  - Skew Handling

What Does Scalable Mean?

- In the past: Out-of-core
  - **"Works even if data doesn't fit in main memory"**
- Now: Parallel
  - **"Can make use of 1000s of independent computers"**

Taxonomy of Parallel Architectures

Scales to 1000s of computers

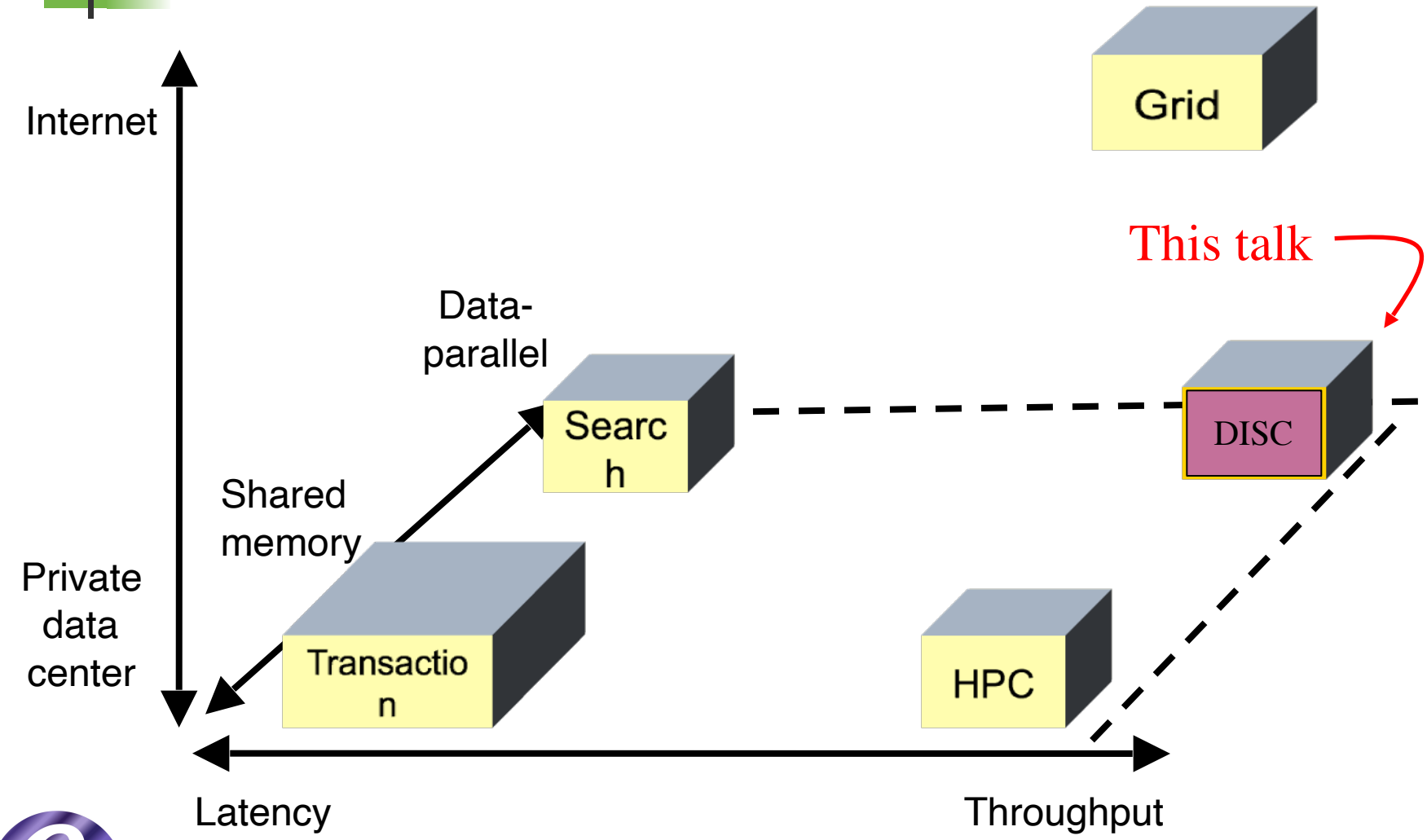


Easiest to program, but

$$$$

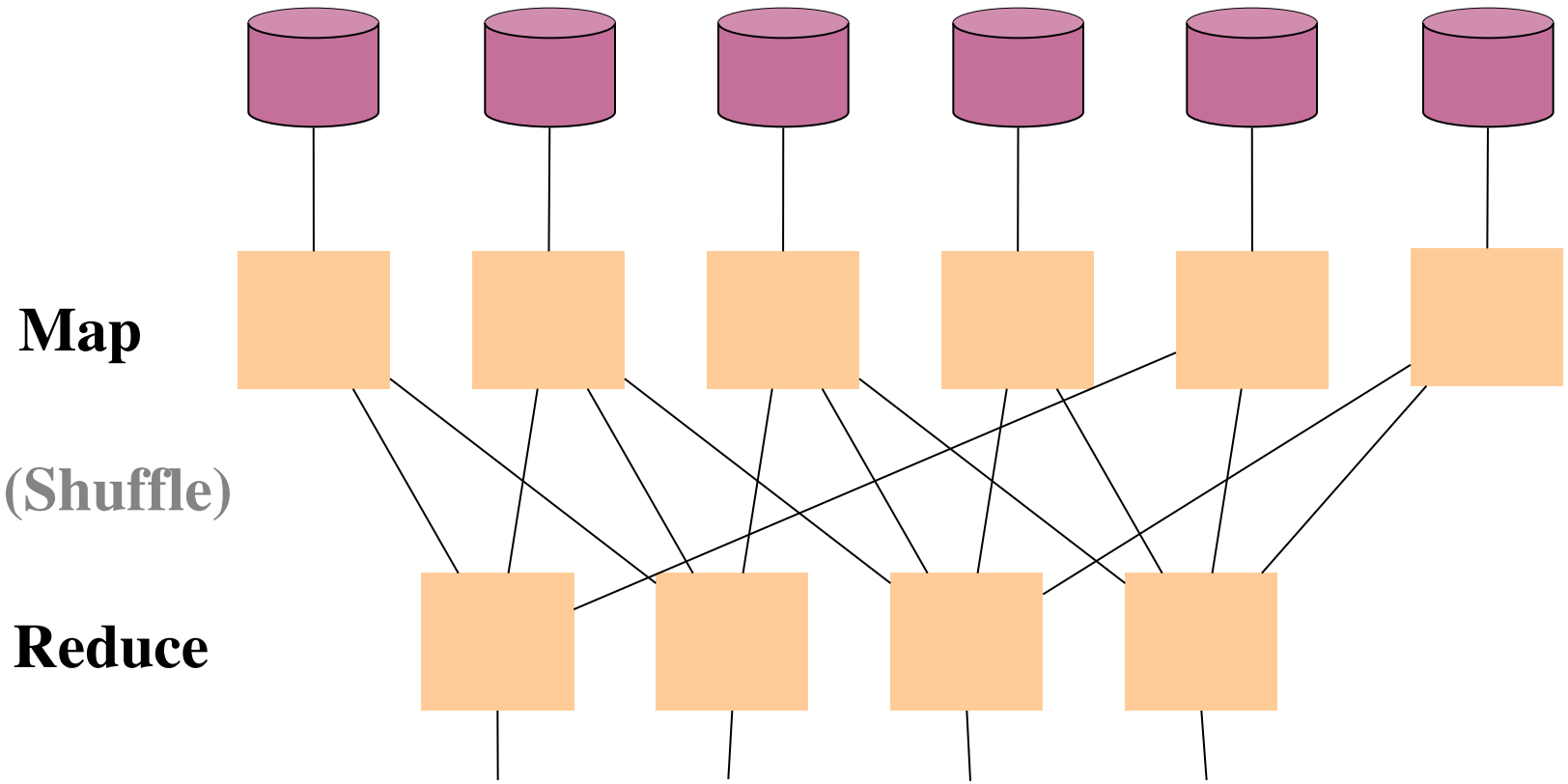
Fig. 3.1 Logical multi-processor database designs (diagram after [DGWI92])

Design Space



slide src: Michael Isard, MSR

Some distributed algorithm…



MapReduce Programming Model

- Input & Output: each a set of key/value pairs
- Programmer specifies two functions:

map (in_key, in_value) -> list(out_key, intermediate_value)

- Processes input key/value pair
- Produces set of intermediate pairs

reduce (out_key, list(intermediate_value)) -> list(out_value)

- Combines all intermediate values for a particular key
- Produces a set of merged output values (usually just one)

Inspired by primitives from functional programming languages such as Lisp, Scheme, and Haskell

slide source: Google, Inc.

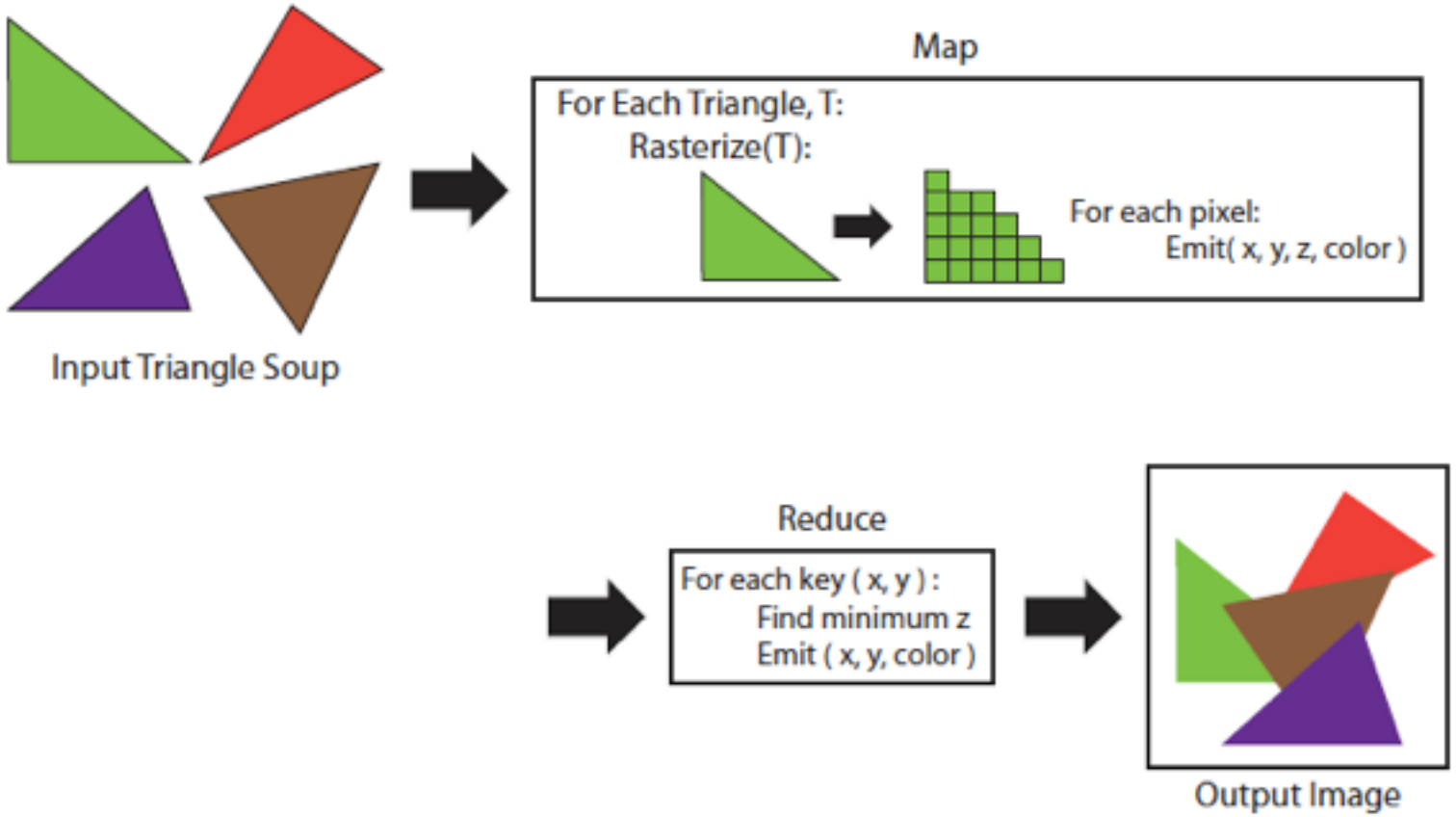
Example: What does this do?

```
map(String input_key, String input_value):
  // input_key: document name
  // input_value: document contents
   for each word w in input_value:
       EmitIntermediate(w, 1);


reduce(String output_key, Iterator intermediate_values):
  // output_key: word
  // output_values: ????
   int result = 0;
   for each v in intermediate_values:
       result += v;
   Emit(result);
```

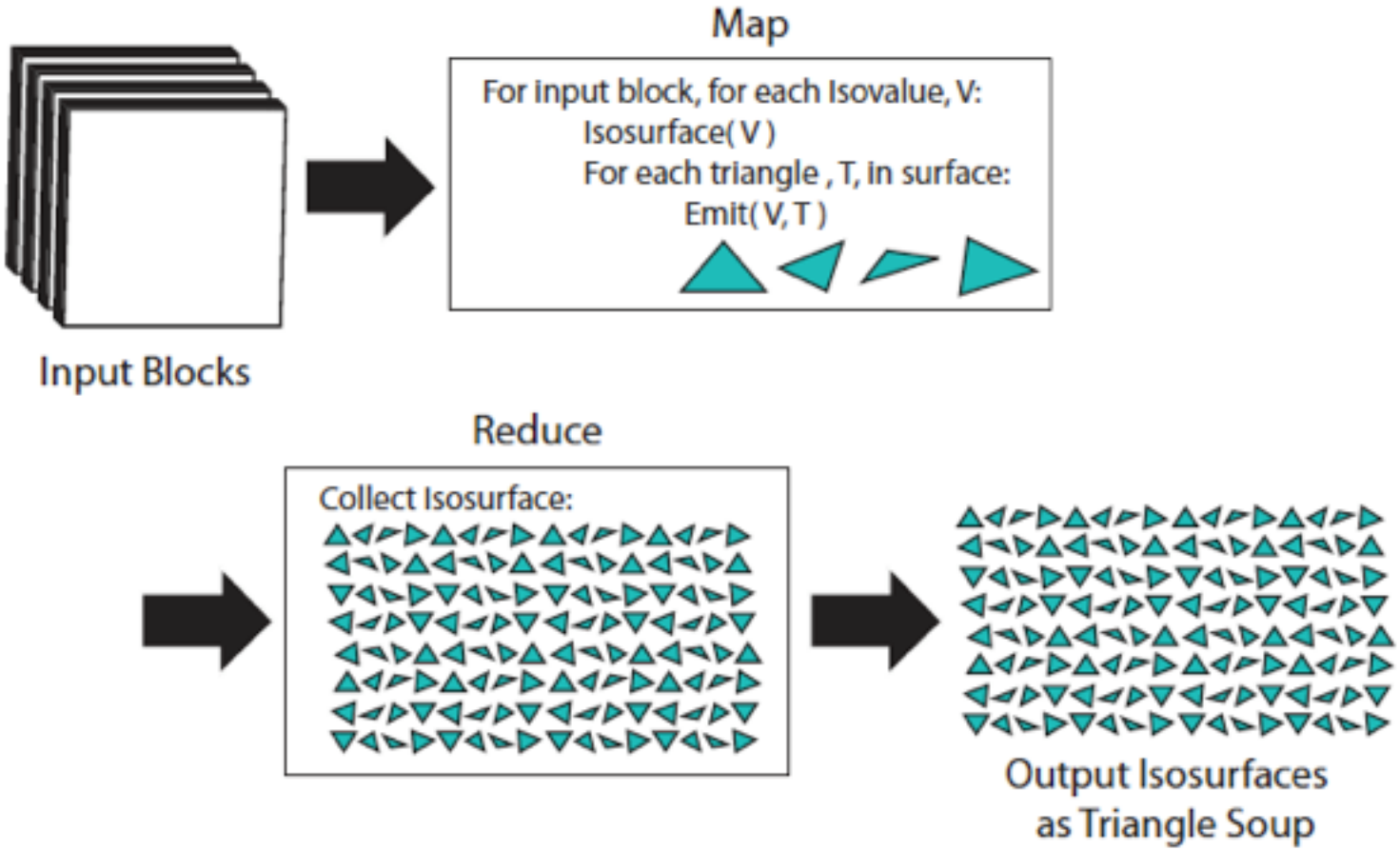
slide source: Google, Inc.

Example: Rendering



Bronson et al. Vis 2010 (submitted)

Example: Isosurface Extraction



Bronson et al. Vis 2010 (submitted)

Large-Scale Data Processing

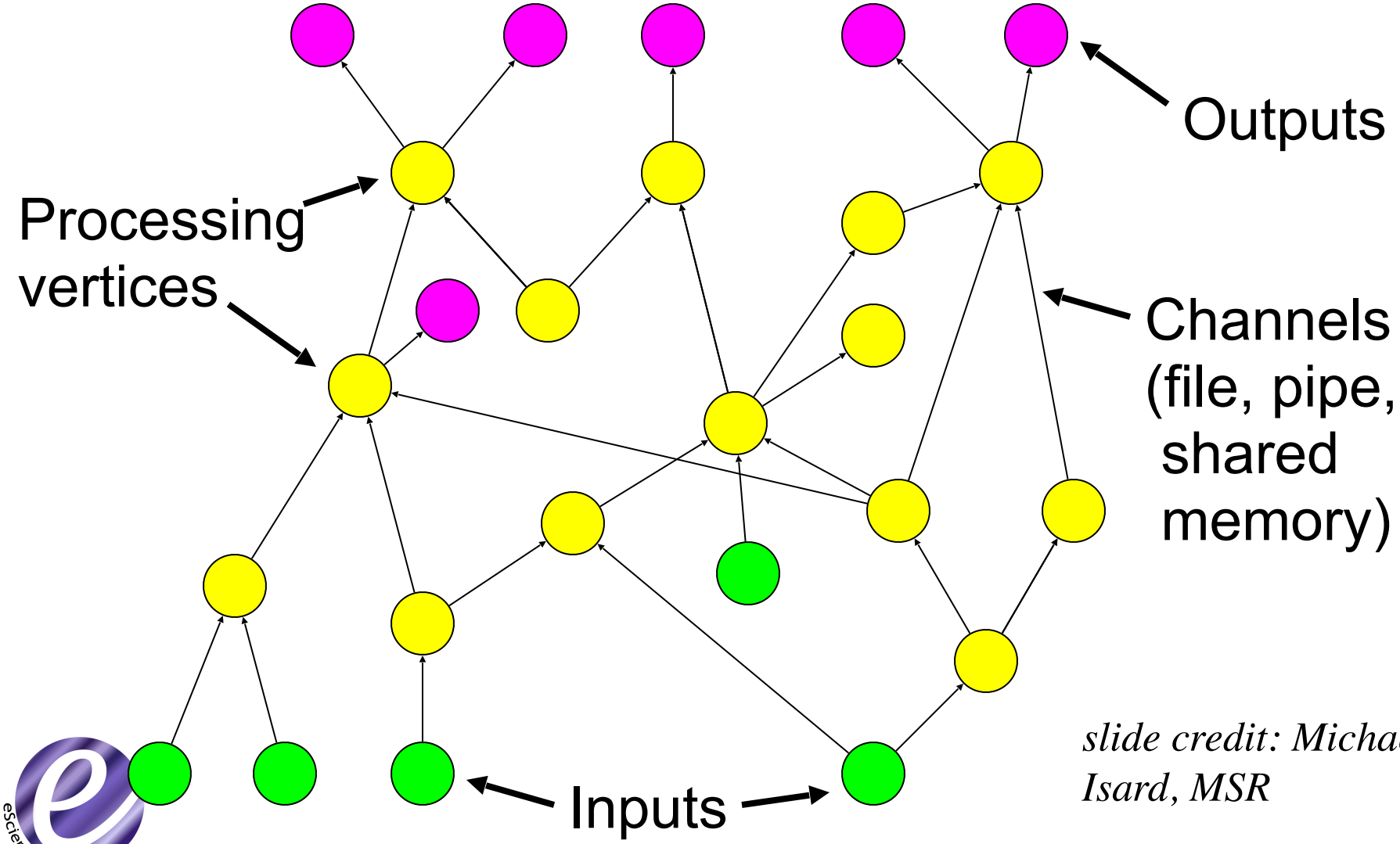
- Many tasks process big data, produce big data
- Want to use hundreds or thousands of CPUs
  - ... but this needs to be easy
  - Parallel databases exist, but they are expensive, difficult to set up, and do not necessarily scale to hundreds of nodes.

- MapReduce is a lightweight framework, providing:
  - **Automatic parallelization and distribution**
  - **Fault-tolerance**
  - **I/O scheduling**
  - **Status and monitoring**

What's wrong with MapReduce?

- Literally Map then Reduce and that's it…
  - Reducers write to replicated storage
- Complex jobs pipeline multiple stages
  - No fault tolerance between stages
    - Map assumes its data is always available: simple!
- What else?

Realistic Job = Directed Acyclic Graph



slide credit: Michael Isard, MSR

Relational Database History

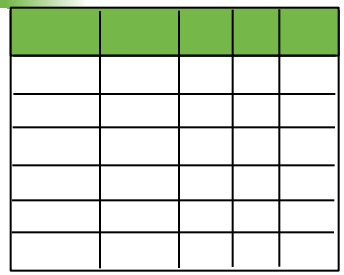
Pre-Relational: if your data changed, your application broke.

Early RDBMS were buggy and slow (and often reviled), but required only 5% of the application code.

> *"Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed."* *-- Codd 1979*

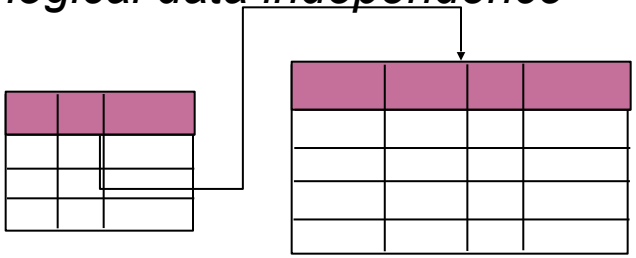
Key Ideas: Programs that manipulate tabular data exhibit an algebraic structure allowing reasoning and manipulation independently of physical data representation

Key Idea: Data Independence



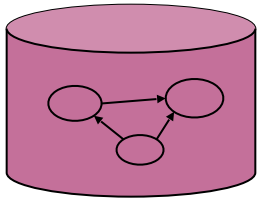
views

logical data independence



relations

physical data independence



files and pointers

```
SELECT *
  FROM my_sequences
```

```
SELECT seq
  FROM ncbi_sequences
 WHERE seq =
'GATTACGATATTA';
```

```
f = fopen('table_file');
fseek(10030440);
while (True) {
  fread(&buf, 1, 8192, f);
  if (buf == GATTACGATATTA) {
    . . .
```

Key Idea: Indexes

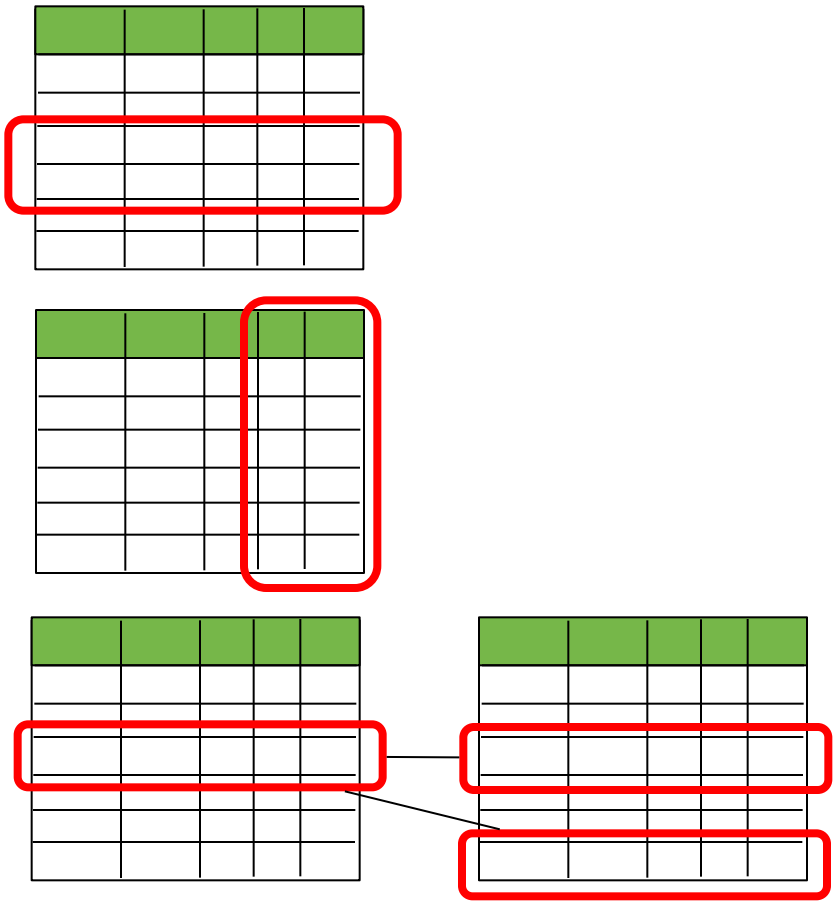
- Databases are especially, but exclusively, effective at “Needle in Haystack” problems:
  - **Extracting small results from big datasets**
  - **Transparently provide “old style” scalability**
  - **Your query will always* finish, regardless of dataset size.**
  - **Indexes are easily built and automatically used when appropriate**

```
CREATE INDEX seq_idx ON sequence(seq);

SELECT seq
  FROM sequence
 WHERE seq = 'GATTACGATATTA';
```

*almost

Key Idea: An *Algebra of Tables*



select

project

join

Other operators: aggregate, union, difference, cross product

Key Idea: Algebraic Optimization

N = ((z*2)+((z*3)+0))/1

Algebraic Laws:

1. (+) identity: x+0 = x
2. (/) identity: x/1 = x
3. (*) distributes: (n*x+n*y) = n*(x+y)
4. (*) commutes: x*y = y*x

Apply rules 1, 3, 4, 2:
N = (2+3)*z

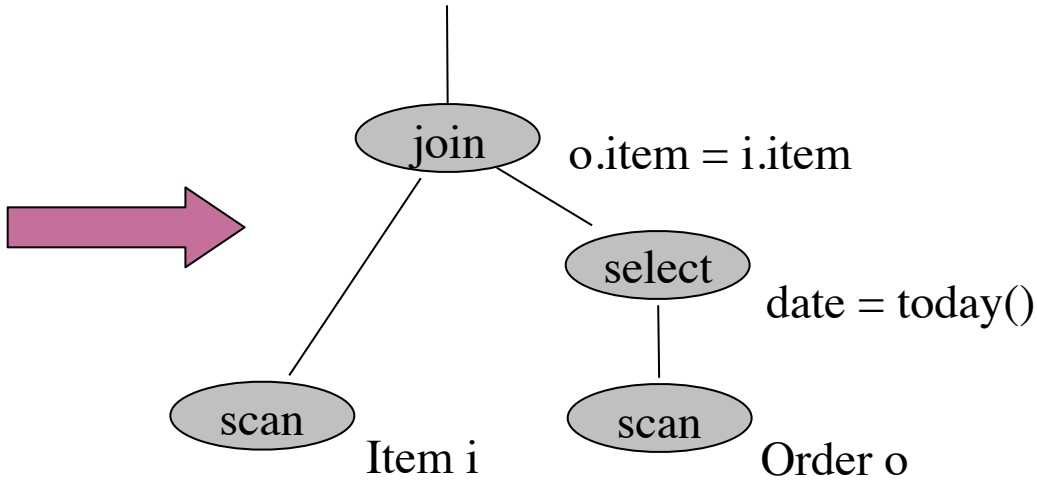
two operations instead of five, no division operator

Same idea works with the Relational Algebra!

Key Idea: Declarative Languages

Find all orders from today, along with the items ordered

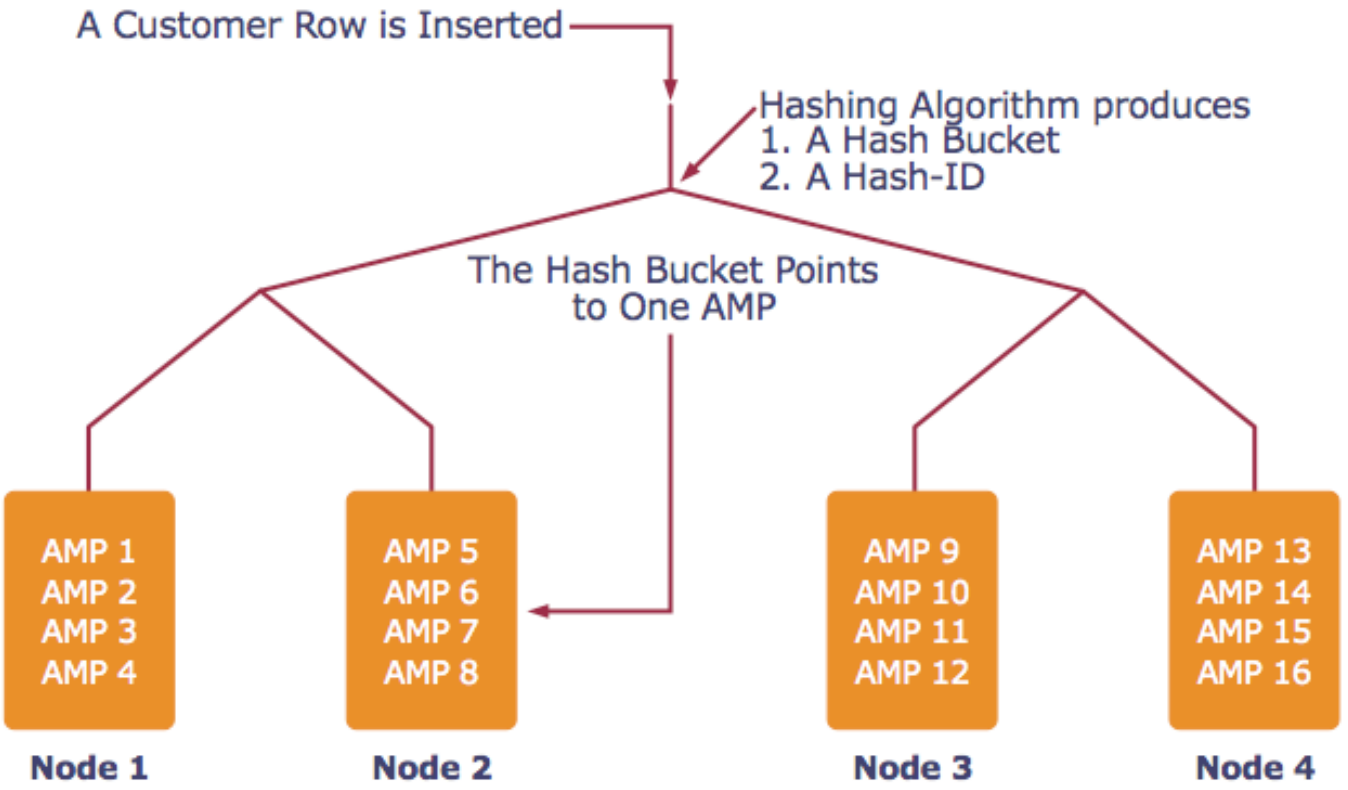
```
SELECT *
  FROM Order o, Item i
 WHERE o.item = i.item
   AND o.date = today()
```



Shared Nothing Parallel Databases

- Teradata
- Greenplum
- Netezza
- Aster Data Systems
- ~~Datallegro~~ **Microsoft**
- Vertica
- MonetDB **Recently commercialized as "Vectorwise"**

Example System: Teradata

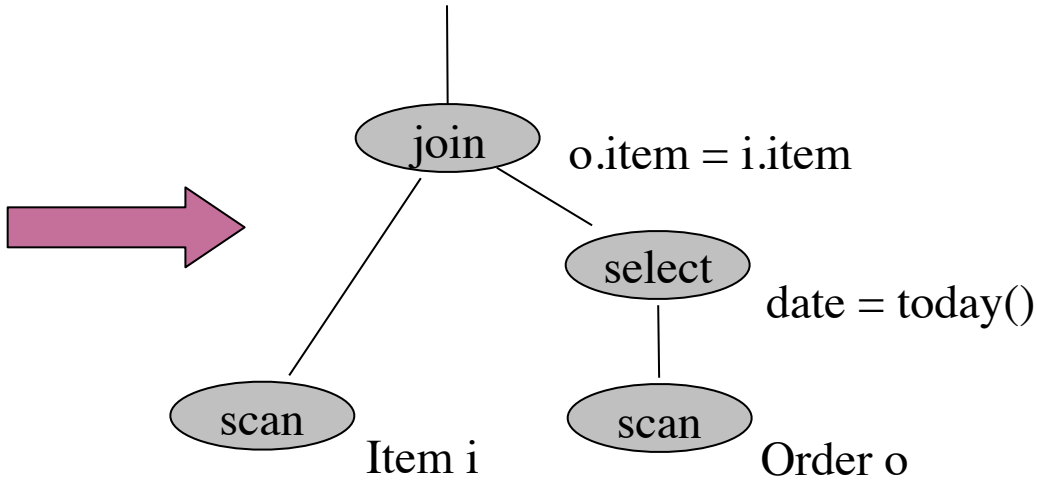


AMP = unit of parallelism

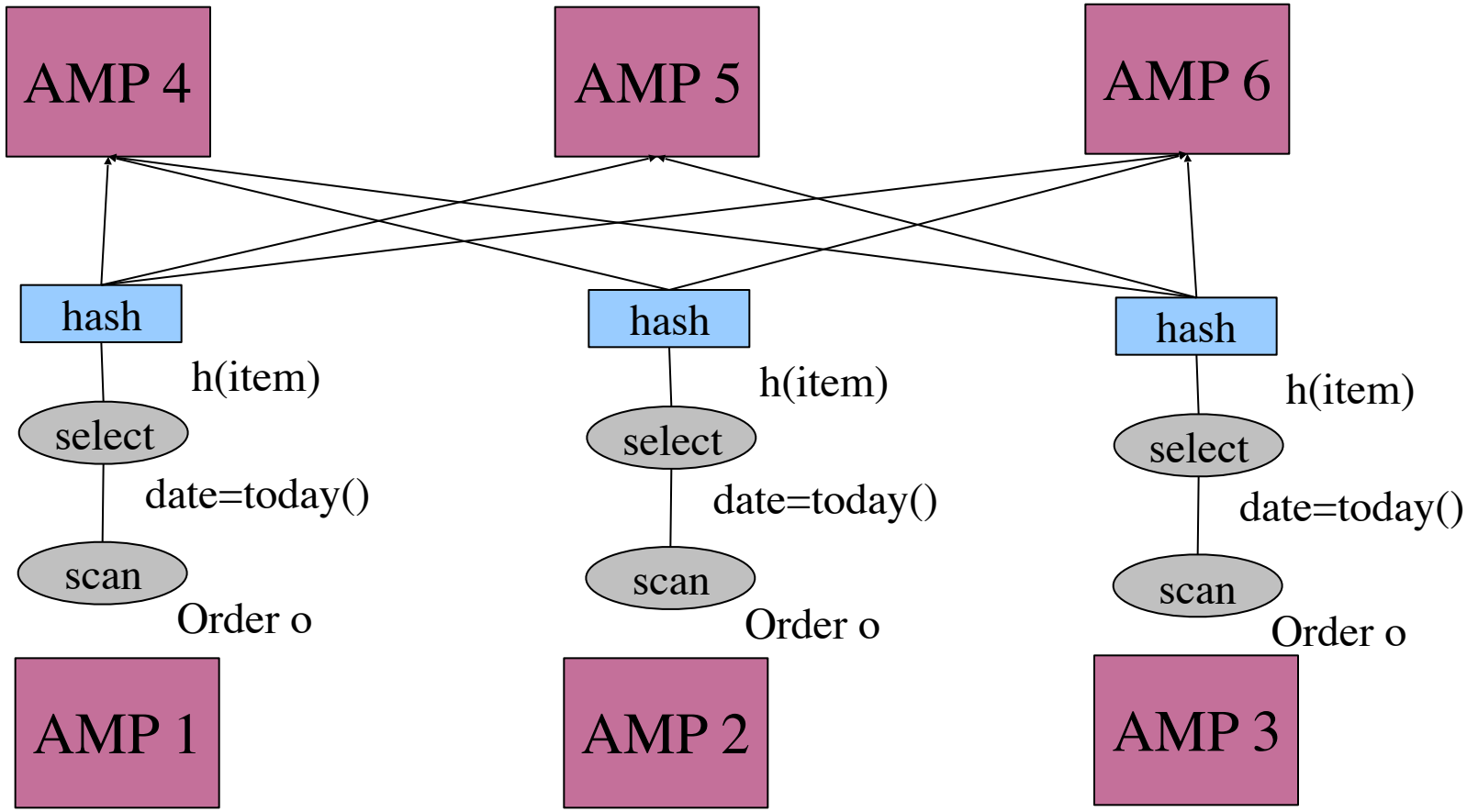
Example System: Teradata

Find all orders from today, along with the items ordered

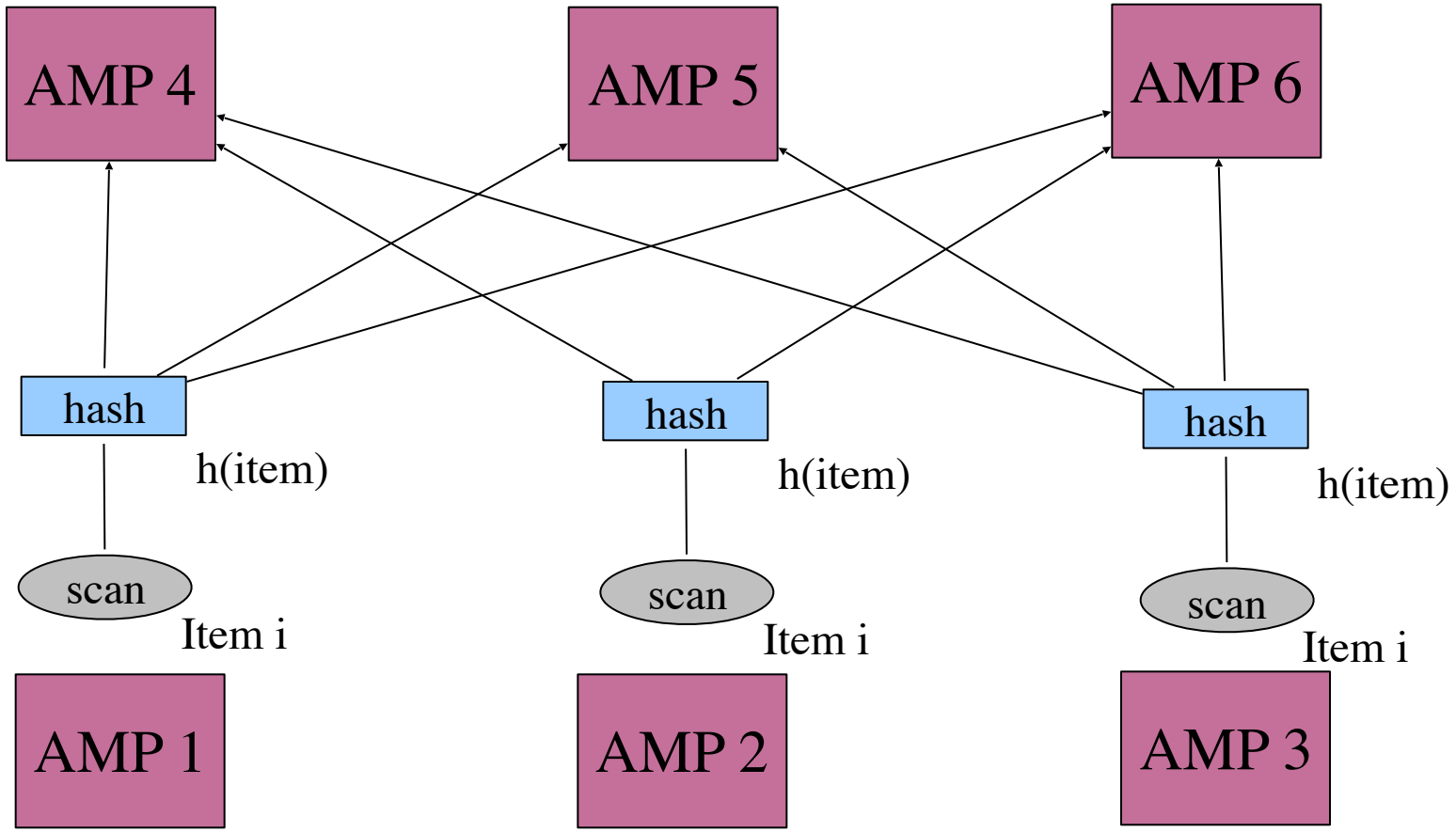
```
SELECT *
  FROM Orders o, Lines i
 WHERE o.item = i.item
   AND o.date = today()
```



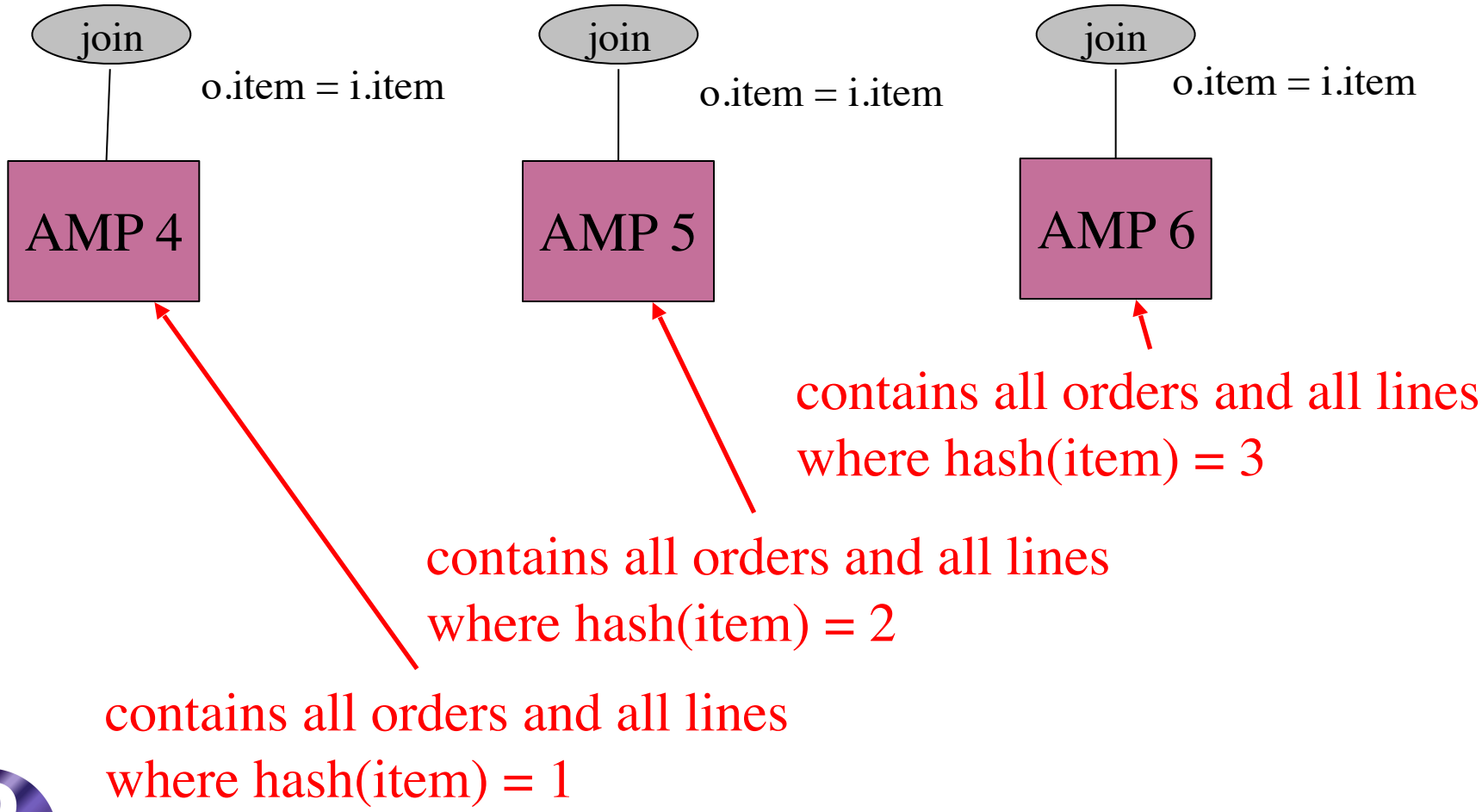
Example System: Teradata



Example System: Teradata



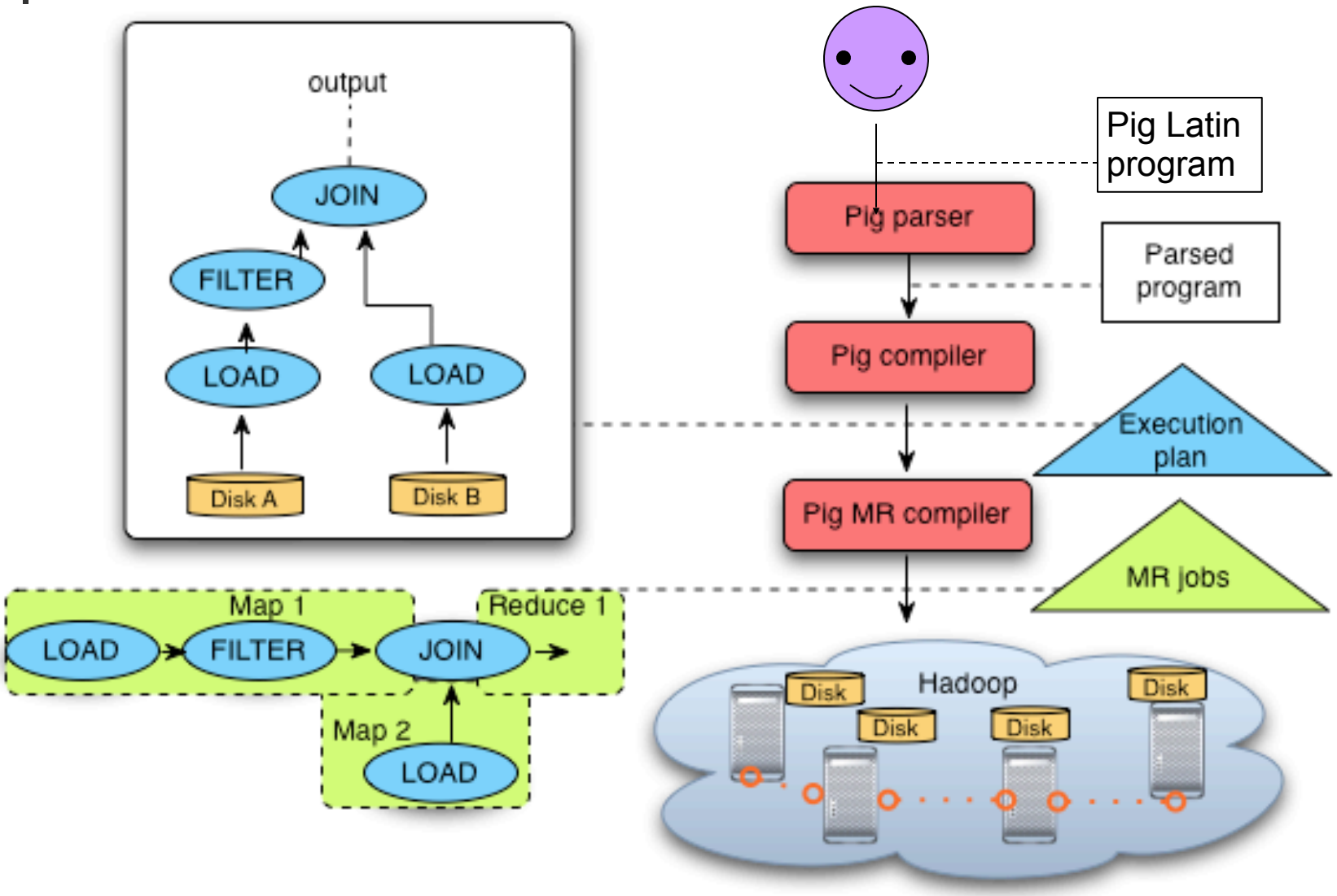
Example System: Teradata



MapReduce Contemporaries

- Dryad (Microsoft)
  - Relational Algebra
- Pig (Yahoo)
  - Near Relational Algebra over MapReduce
- HIVE (Facebook)
  - SQL over MapReduce
- Cascading
  - Relational Algebra
- Clustera
  - U of Wisconsin
- Hbase
  - Indexing on HDFS

Example System: Yahoo Pig



MapReduce vs RDBMS

- RDBMS
  - Declarative query languages
  - Schemas
  - Logical Data Independence
  - Indexing
  - Algebraic Optimization
  - Caching/Materialized Views
  - *ACID/Transactions*
- MapReduce
  - High Scalability
  - Fault-tolerance
  - "One-person deployment"

Dryad, Pig, HIVE
HIVE, Pig, Dryad

Hbase
Pig, (Dryad, HIVE)

	Data Model	Prog. Model	Services
GPL	*	*	Typing (maybe)
Workflow	*	dataflow	typing, provenance, scheduling, caching, task parallelism, reuse
Relational Algebra	Relations	Select, Project, Join, Aggregate, …	optimization, physical data independence, data parallelism
MapReduce	[(key,value)]	Map, Reduce	massive data parallelism, fault tolerance
MS Dryad	IQueryable, IEnumerable	RA + Apply + Partitioning	typing, massive data parallelism, fault tolerance
MPI	Arrays/ Matrices	70+ ops	data parallelism, full control

Roadmap

- Introduction
- Context: RDBMS, MapReduce, etc.
- New Extensions for Science
  - Recursive MapReduce
  - Skew Handling

PageRank

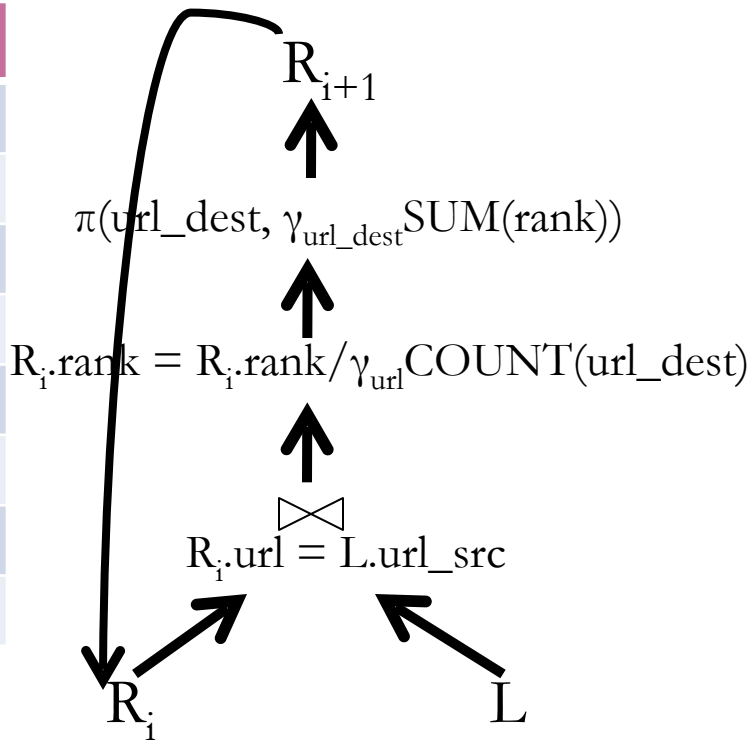
[Bu et al. VLDB 2010 (submitted)]

Rank Table R_0

url	rank
www.a.com	1.0
www.b.com	1.0
www.c.com	1.0
www.d.com	1.0
www.e.com	1.0

Linkage Table *L*

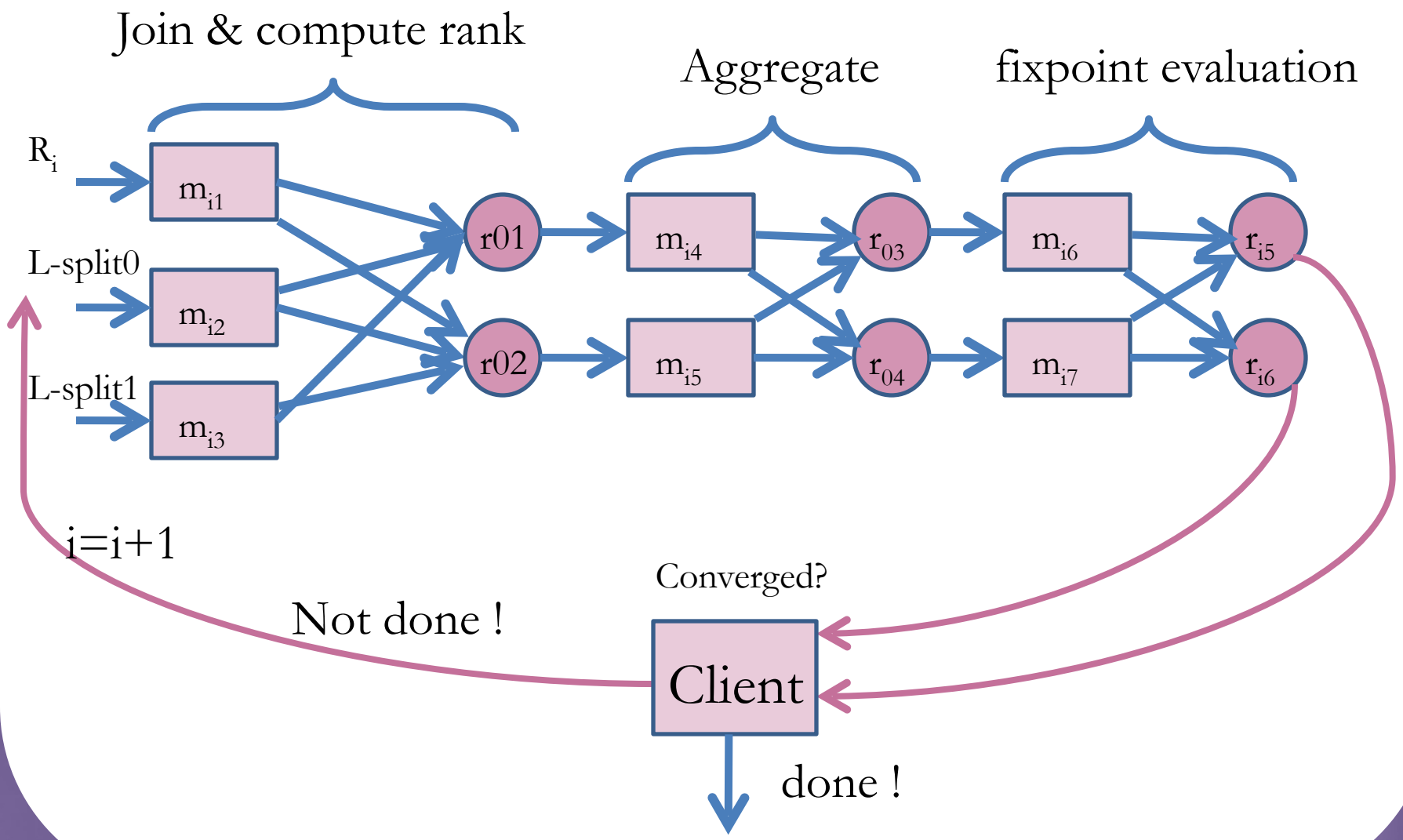
url_src	url_dest
www.a.com	www.b.com
www.a.com	www.c.com
www.c.com	www.a.com
www.e.com	www.c.com
www.d.com	www.b.com
www.c.com	www.e.com
www.e.com	www.c.com
www.a.com	www.d.com



Rank Table R_3

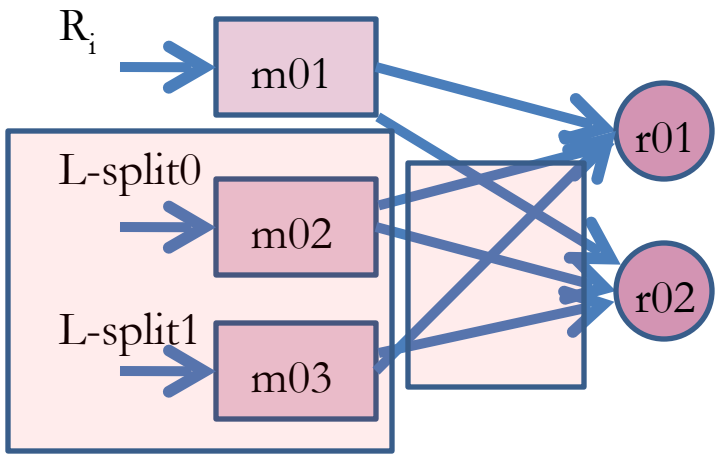
url	rank
www.a.com	2.13
www.b.com	3.89
www.c.com	2.60
www.d.com	2.60
www.e.com	2.13

MapReduce Implementation



[Bu et al. VLDB 2010 (submitted)]

What's the problem?



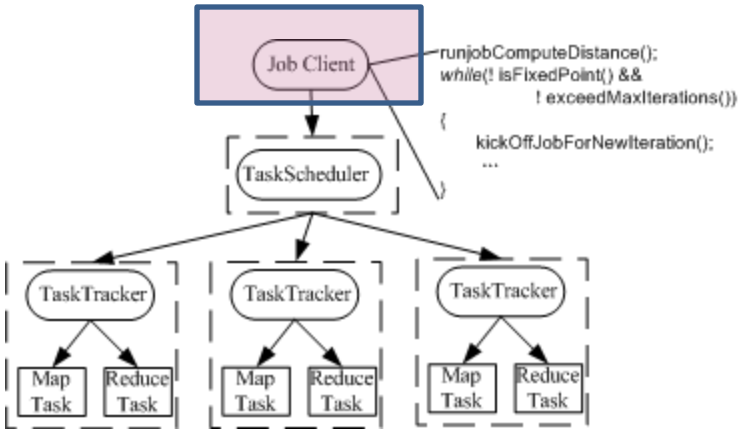
- *L* is loaded and shuffled in each iteration
- *L* never changes

[Bu et al. VLDB 2010 (submitted)]

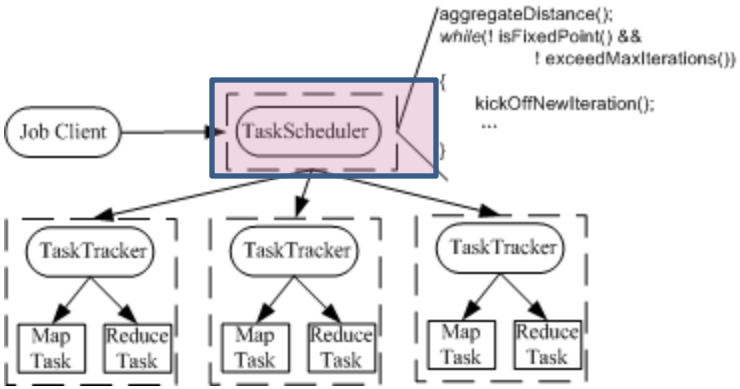
[Bu et al. VLDB 2010 (submitted)]

HaLoop: Loop-aware Hadoop

Hadoop



HaLoop



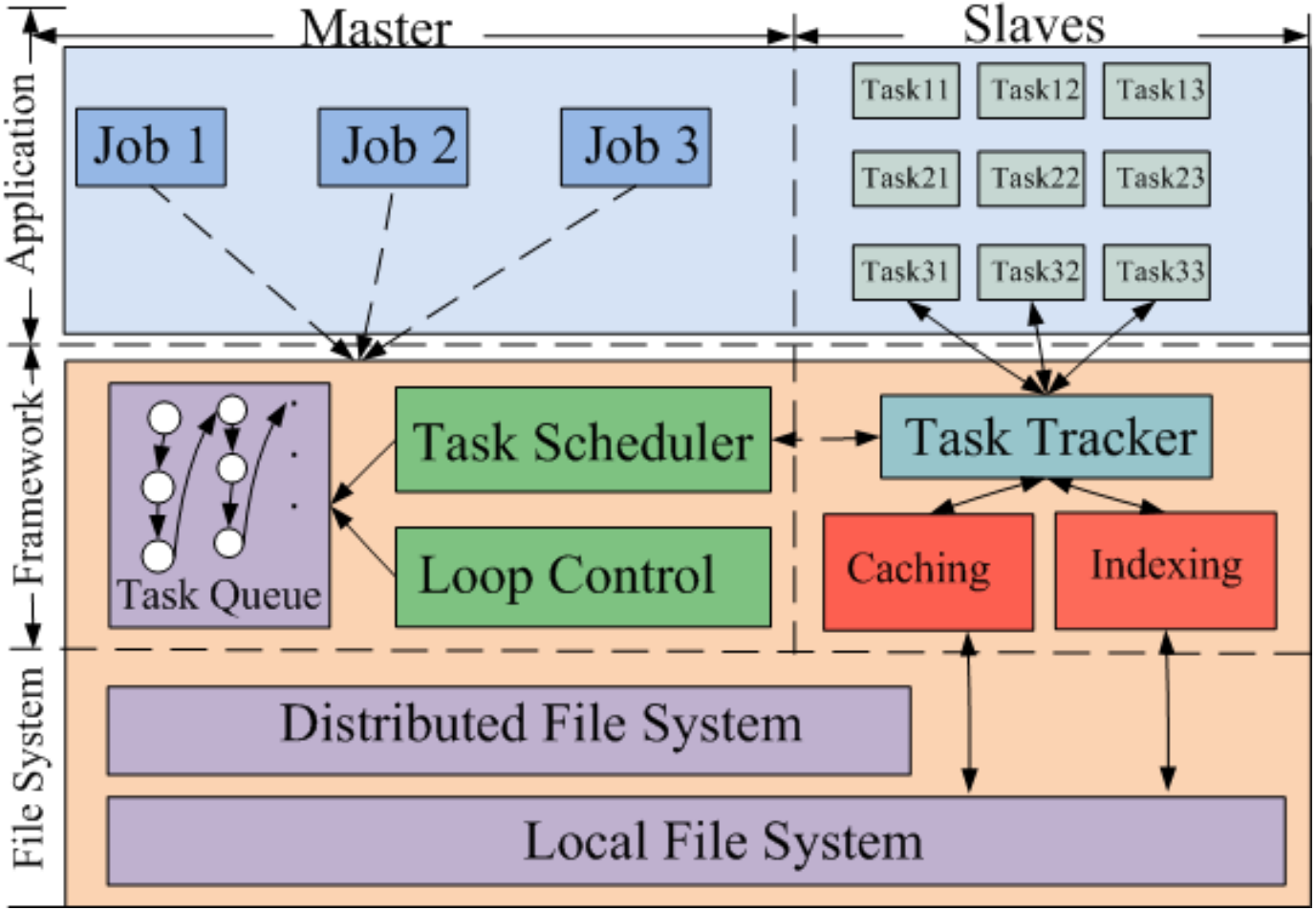
- Hadoop: Loop control in client program
- HaLoop: Loop control in master node

Feature: Inter-iteration Locality

- Mapper Output Cache
  - K-means
  - Neural network analysis
- Reducer Input Cache
  - Recursive join
  - PageRank
  - HITs
  - Social network analysis
- Reducer Output Cache
  - Fixpiont evaluation

[Bu et al. VLDB 2010 (submitted)]

HaLoop Architecture



[Bu et al. VLDB 2010 (submitted)]

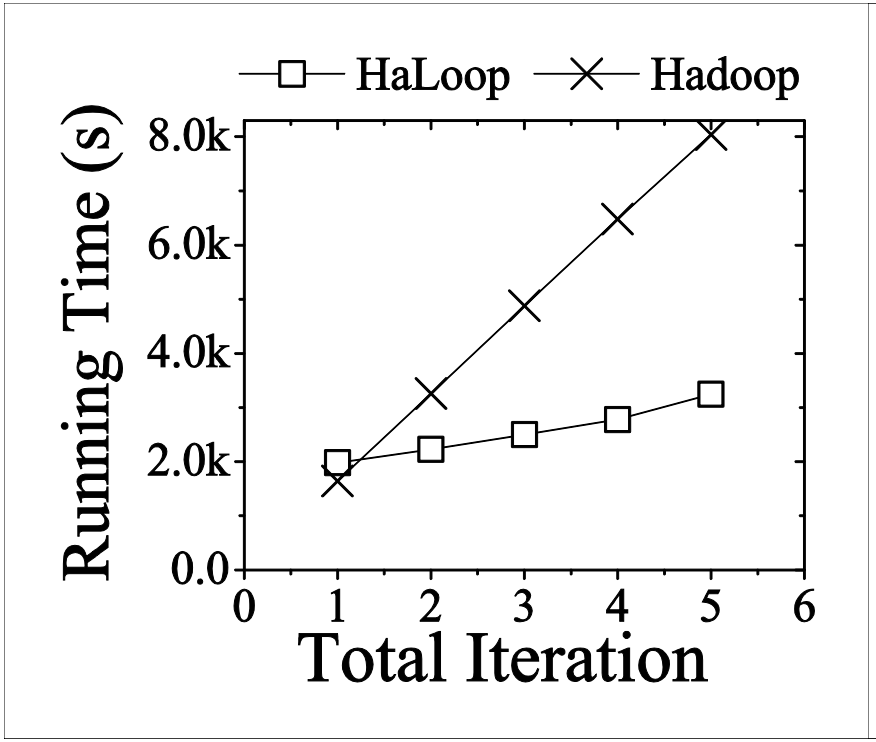
Experiments

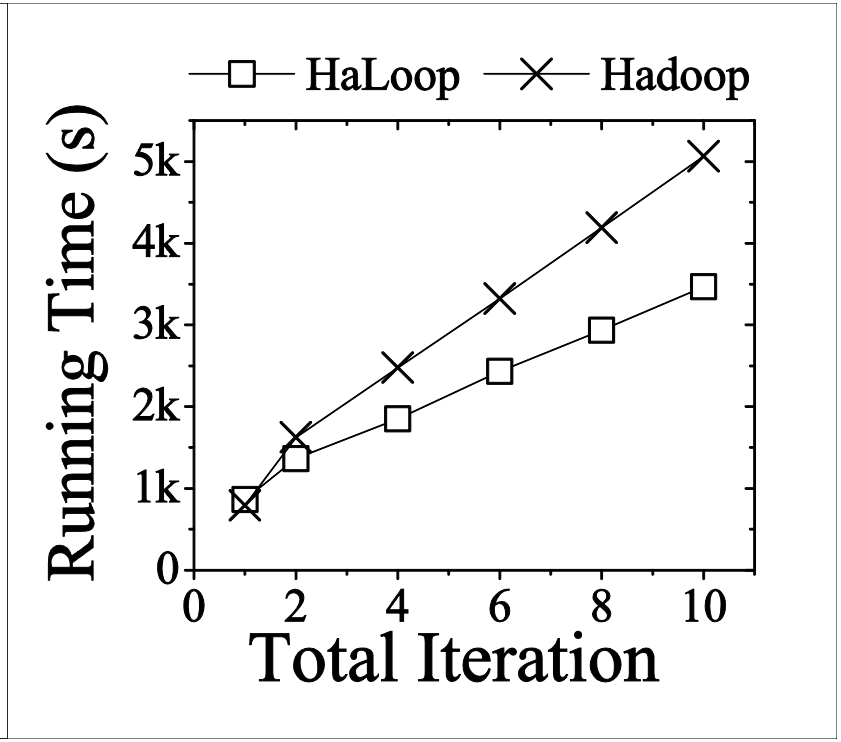
- Amazon EC2
  - 20, 50, 90 default small instances
- Datasets
  - Billions of Triples (120GB)
  - Freebase (12GB)
  - Livejournal social network (18GB)
- Queries
  - Transitive Closure
  - PageRank
  - k-means

[Bu et al. VLDB 2010 (submitted)]

[Bu et al. VLDB 2010 (submitted)]

Application Run Time

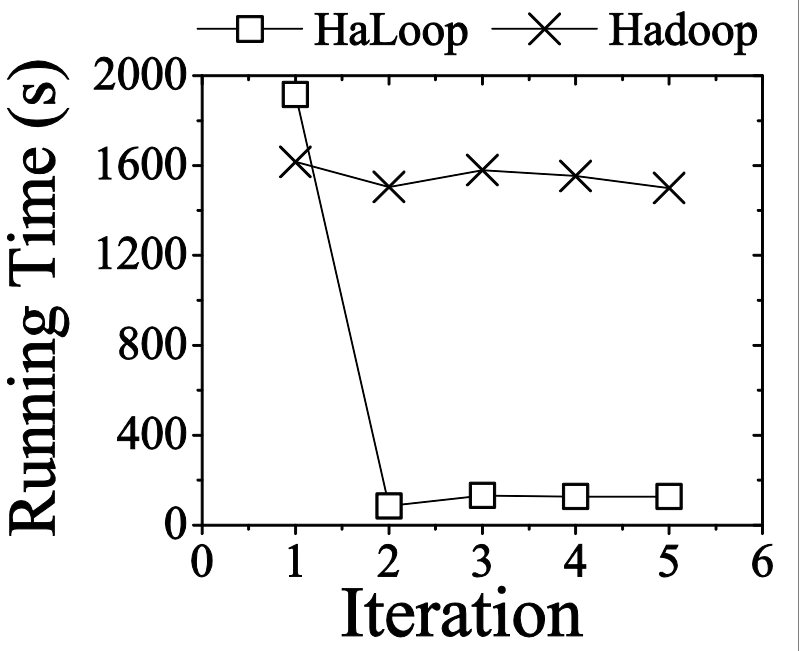


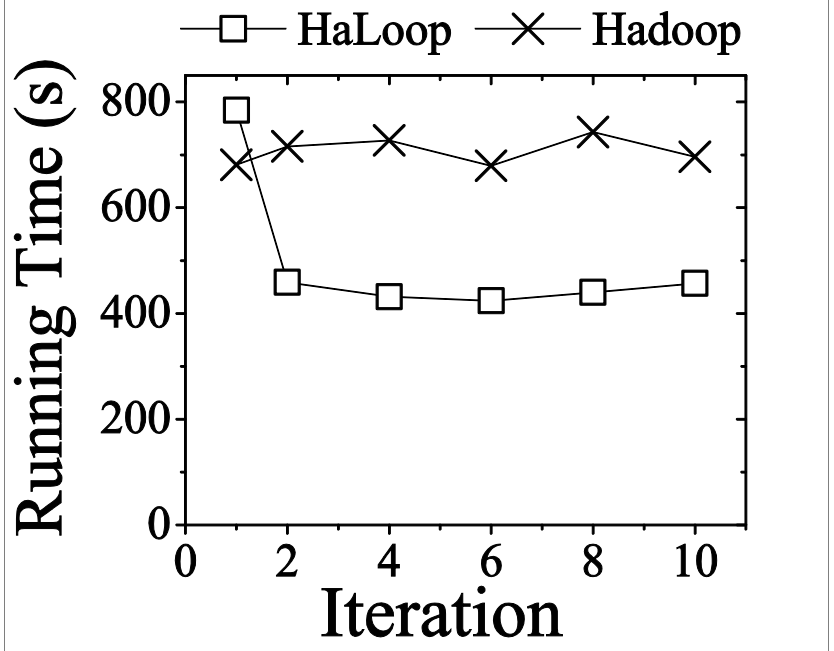


- Transitive Closure
- PageRank

[Bu et al. VLDB 2010 (submitted)]

Join Time

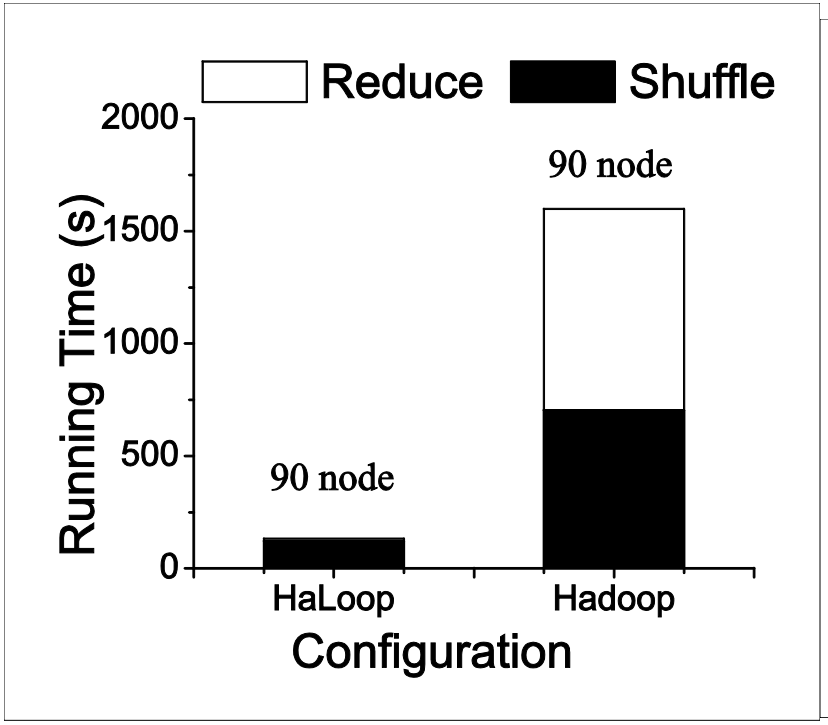


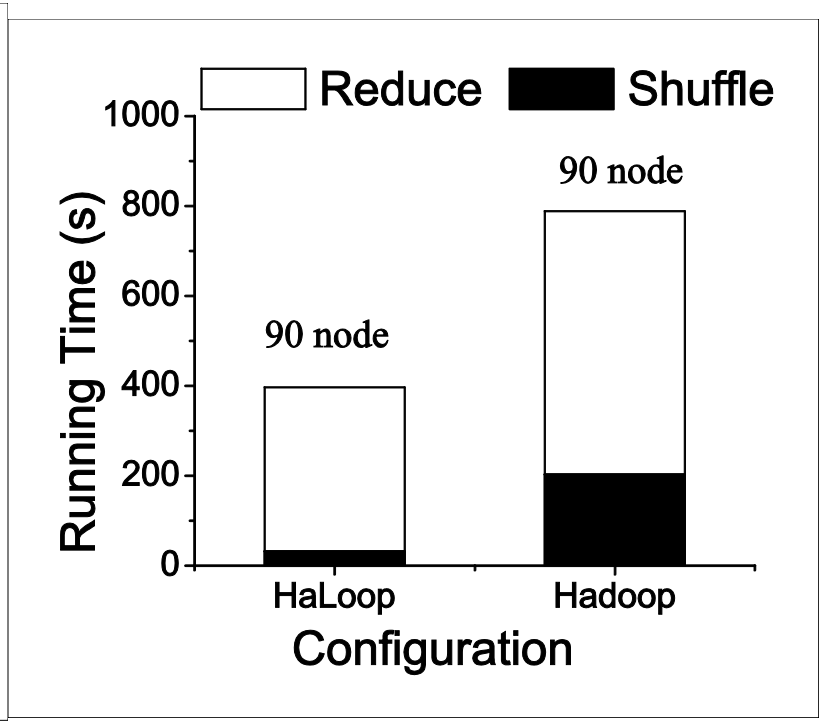


- Transitive Closure
- PageRank

[Bu et al. VLDB 2010 (submitted)]

Run Time Distribution

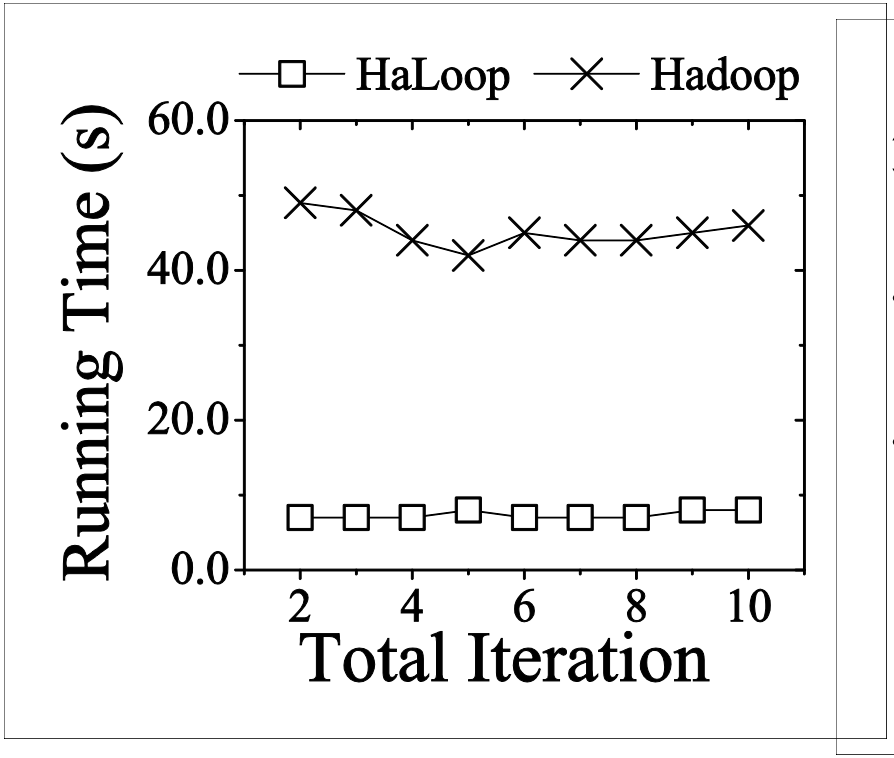


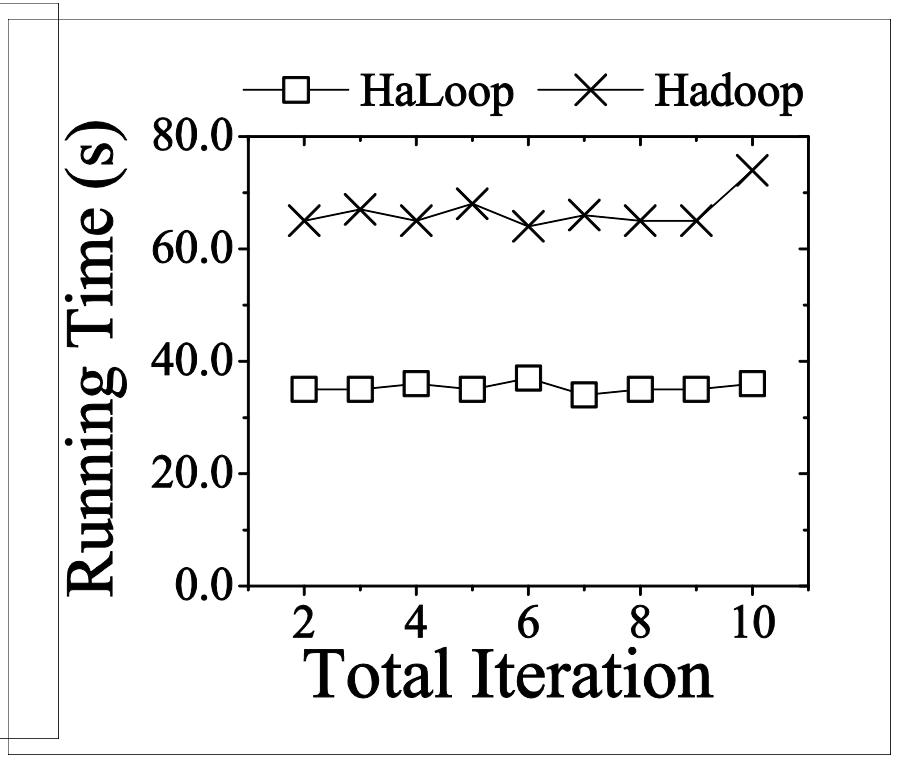


- Transitive Closure
- PageRank

[Bu et al. VLDB 2010 (submitted)]

Fixpoint Evaluation





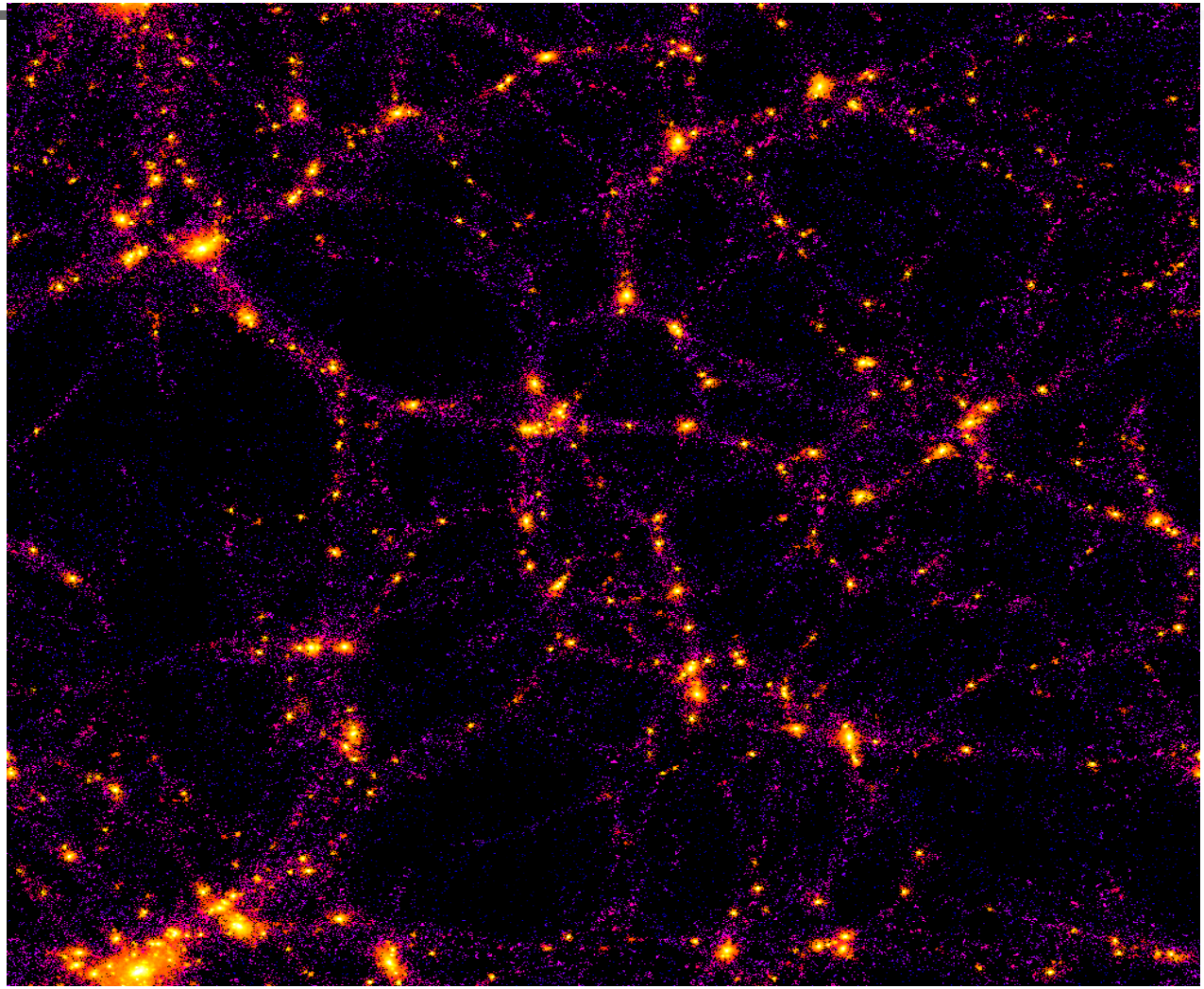
- PageRank

Roadmap

- Introduction
- Context: RDBMS, MapReduce, etc.
- New Extensions for Science
  - Recursive MapReduce
  - Skew Handling

N-body Astrophysics Simulation

- 15 years in dev
- 10^9 particles
- Months to run
- 7.5 million CPU hours
- 500 timesteps
- Big Bang to now



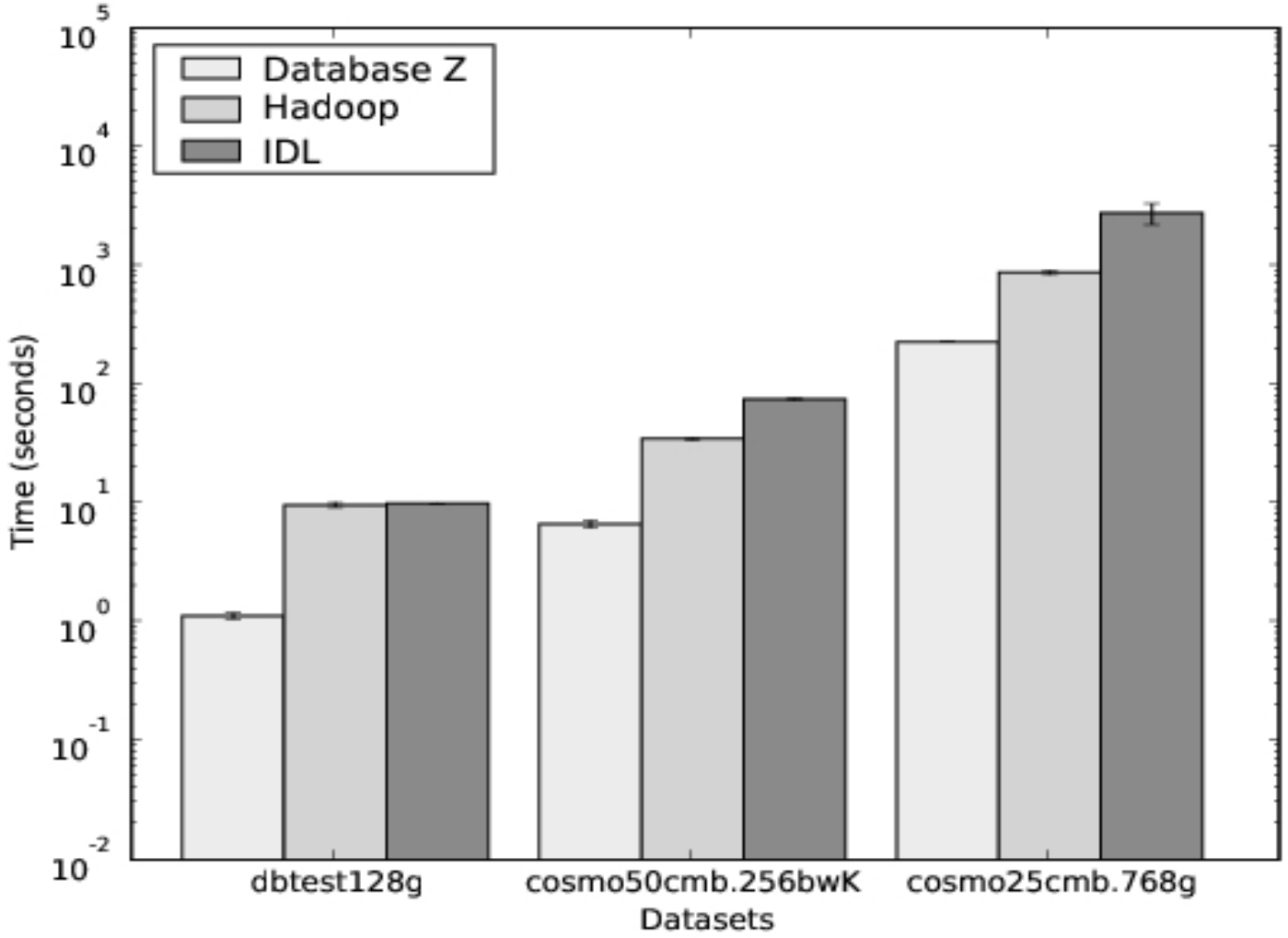
Simulations from Tom Quinn's Lab, work by Sarah Loebman, YongChul Kwon, Bill Howe, Jeff Gardner, Magda Balazinska

Q1: Find Hot Gas

```
SELECT id
  FROM gas
 WHERE temp > 150000
```

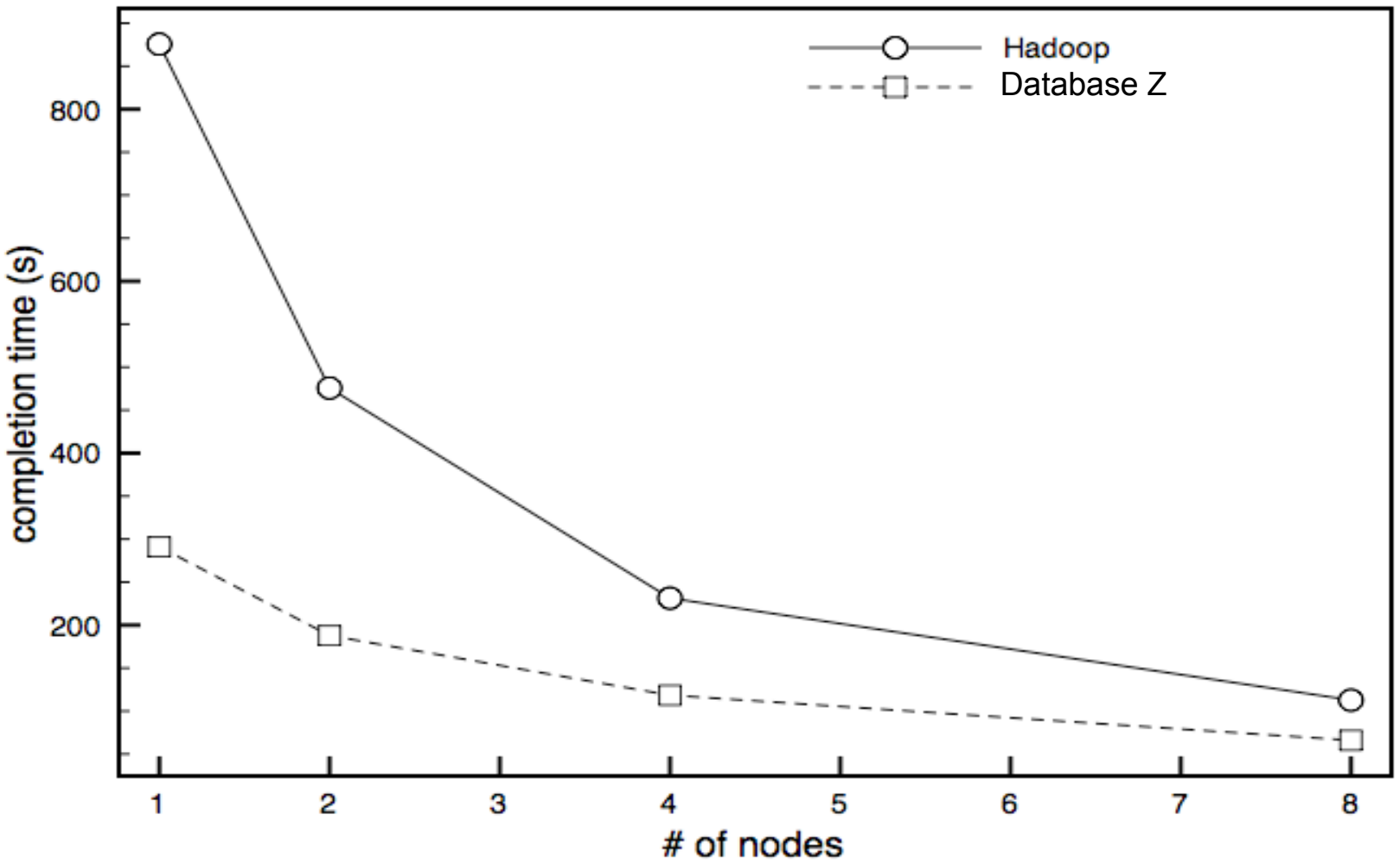
Single Node: Query 1

[IASDS 09]



169 MB 1.4 GB 36 GB

Multiple Nodes: Query 1 [IASDS 09]



Q4: Gas Deletion

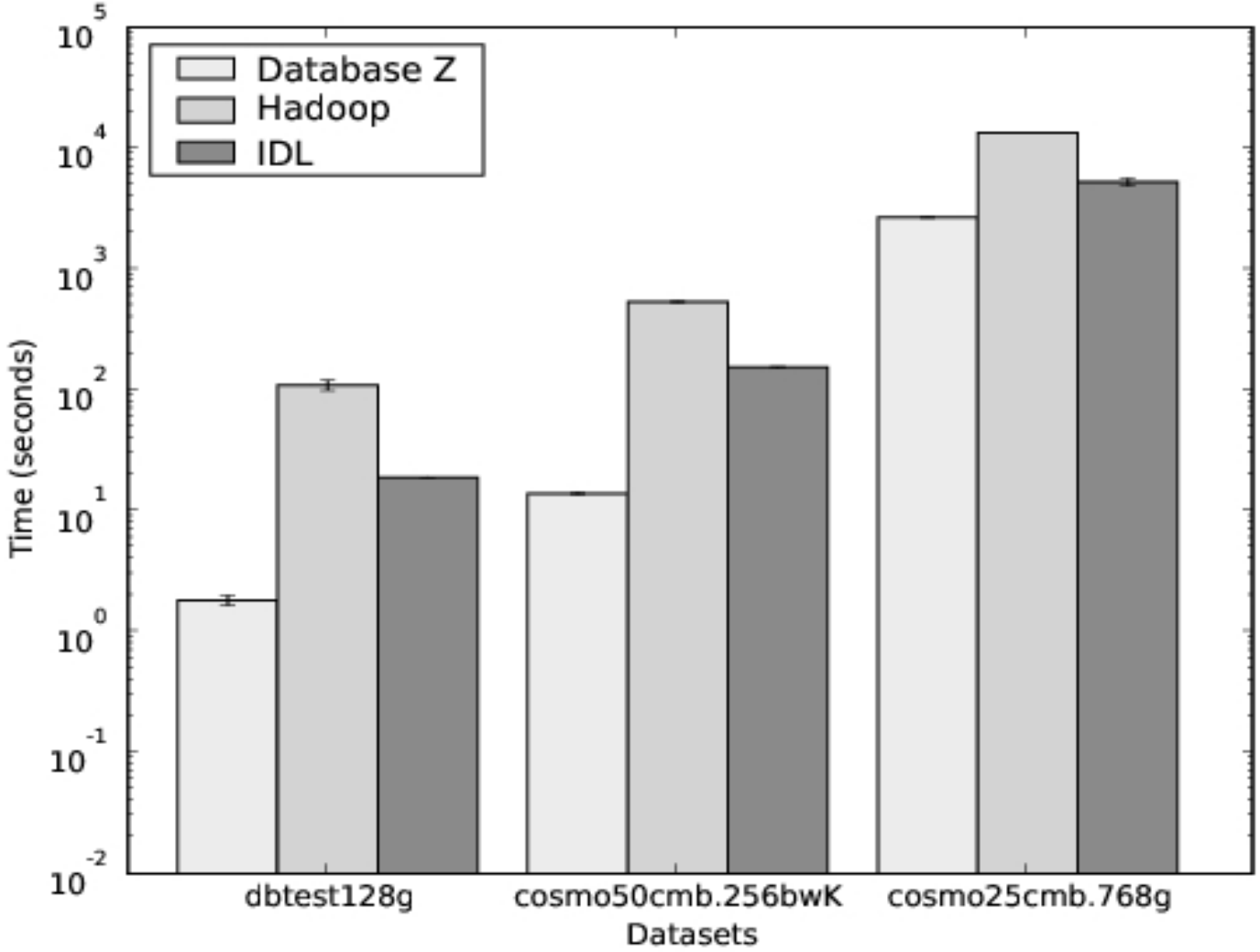
[IASDS 09]

```
SELECT gas1.id
FROM gas1
FULL OUTER JOIN gas2
ON gas1.id=gas2.id
WHERE gas2.id=NULL
```

Particles removed
between two timesteps

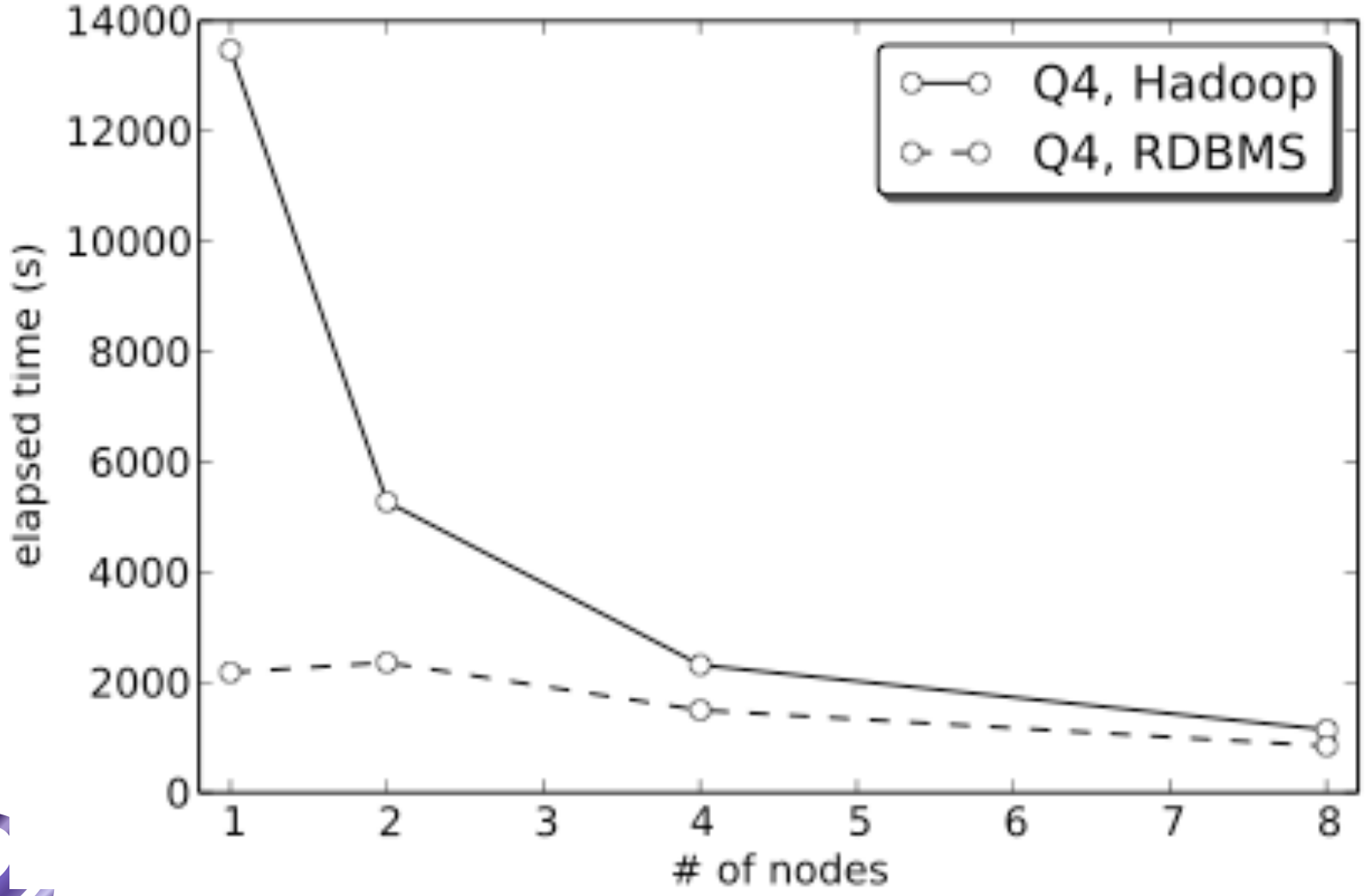
Single Node: Query 4

[IASDS 09]



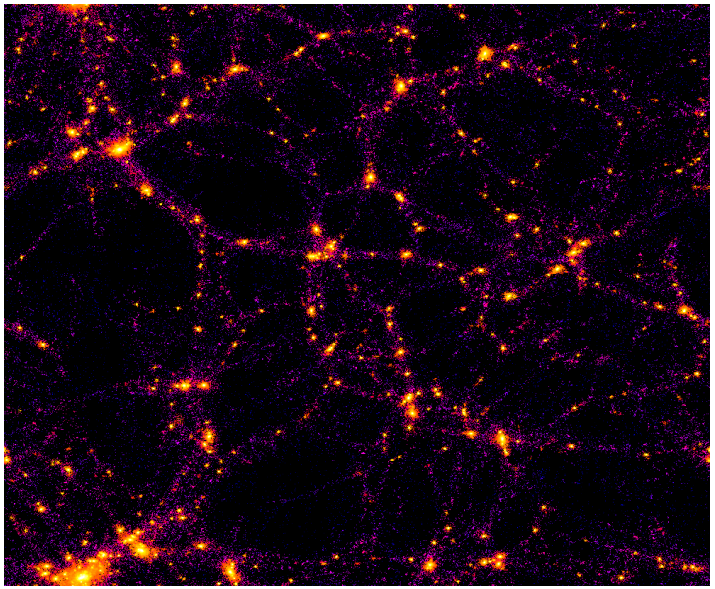
Multiple Nodes: Query 4

[IASDS 09]



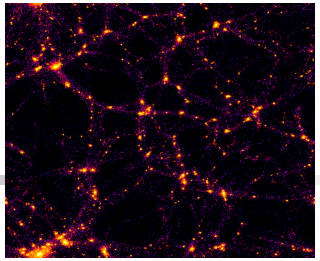
New Task: Scalable Clustering

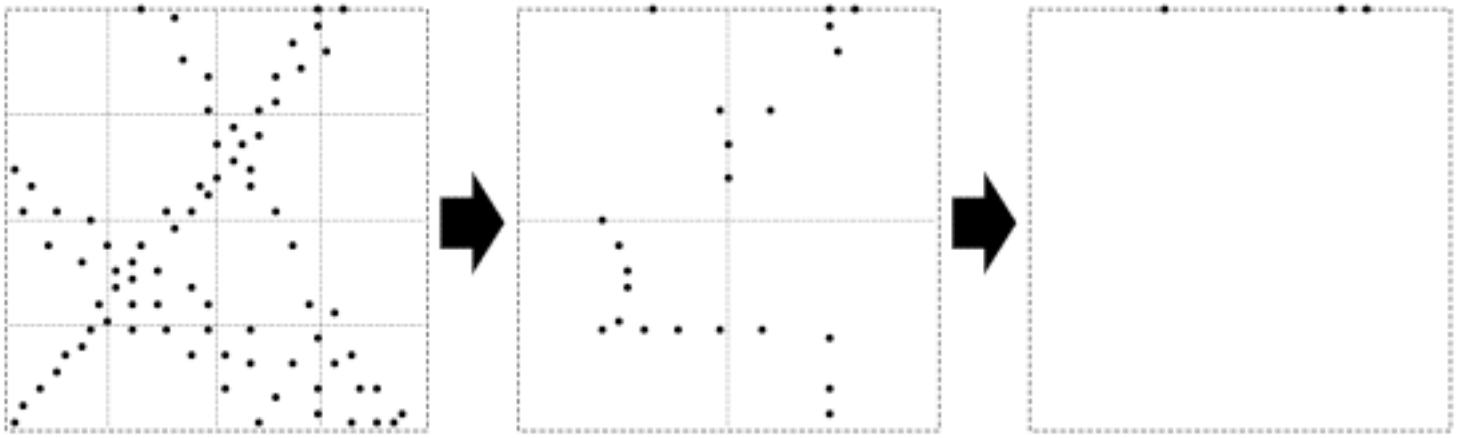
- Group particles into spatial clusters



[Kwon SSDBM 2010]

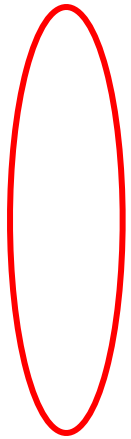
Scalable Clustering





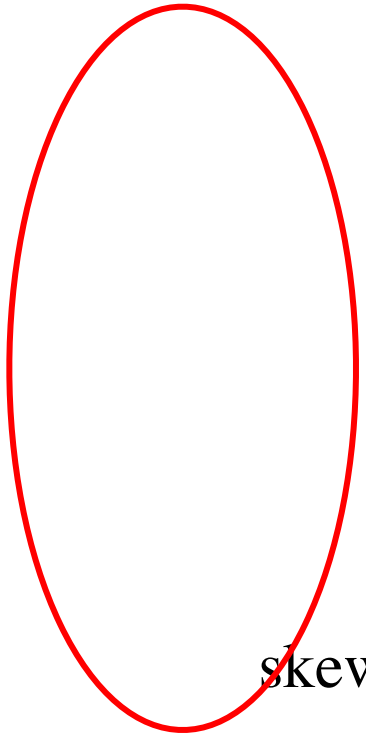
[Kwon SSDBM 2010]

Scalable Clustering in Dryad



[Kwon SSDBM 2010]

Scalable Clustering in Dryad



non-skewed

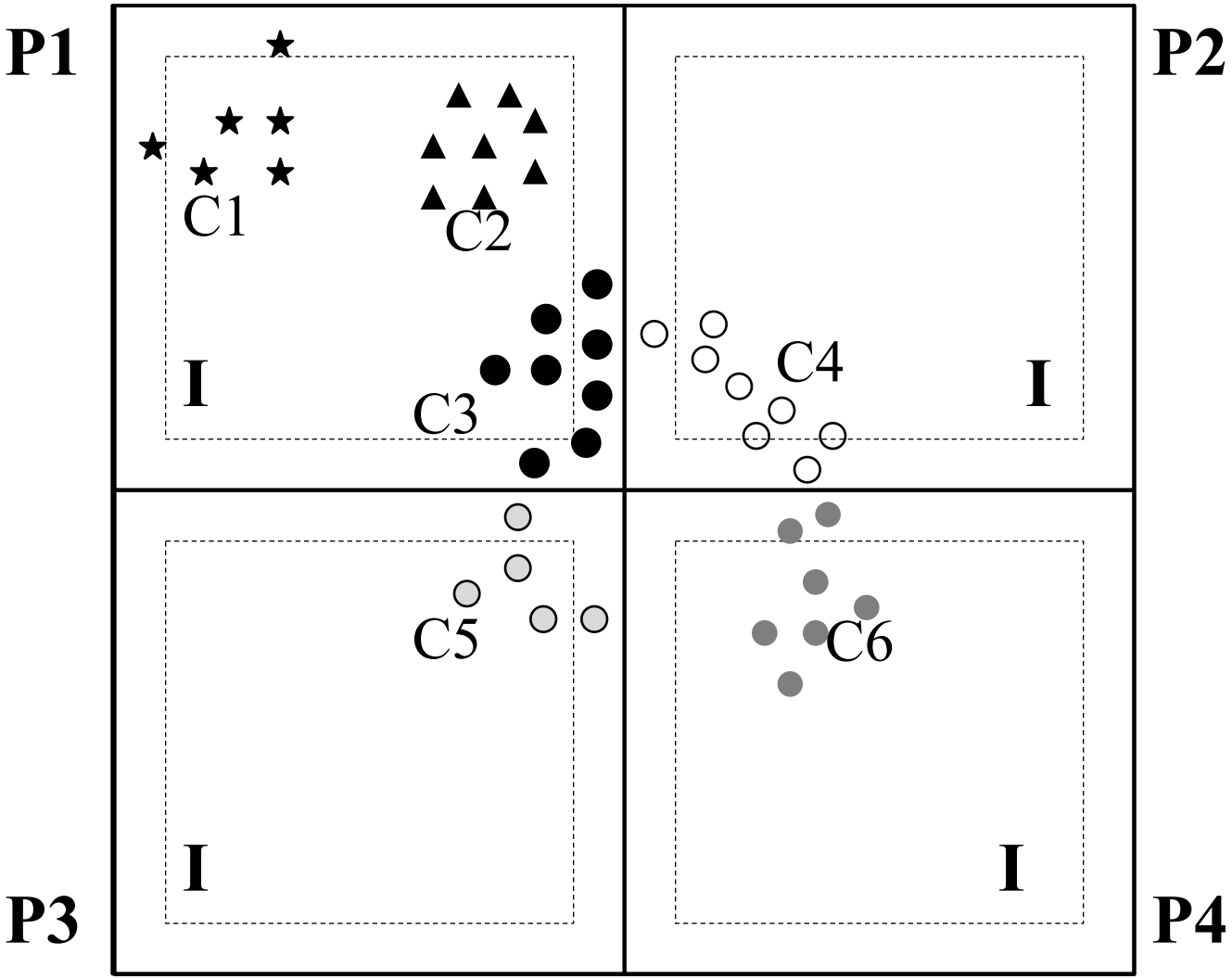
skewed

YongChul Kwon, Dylan Nunlee, Jeff Gardner, Sarah Loebman, Magda Balazinska, Bill Howe

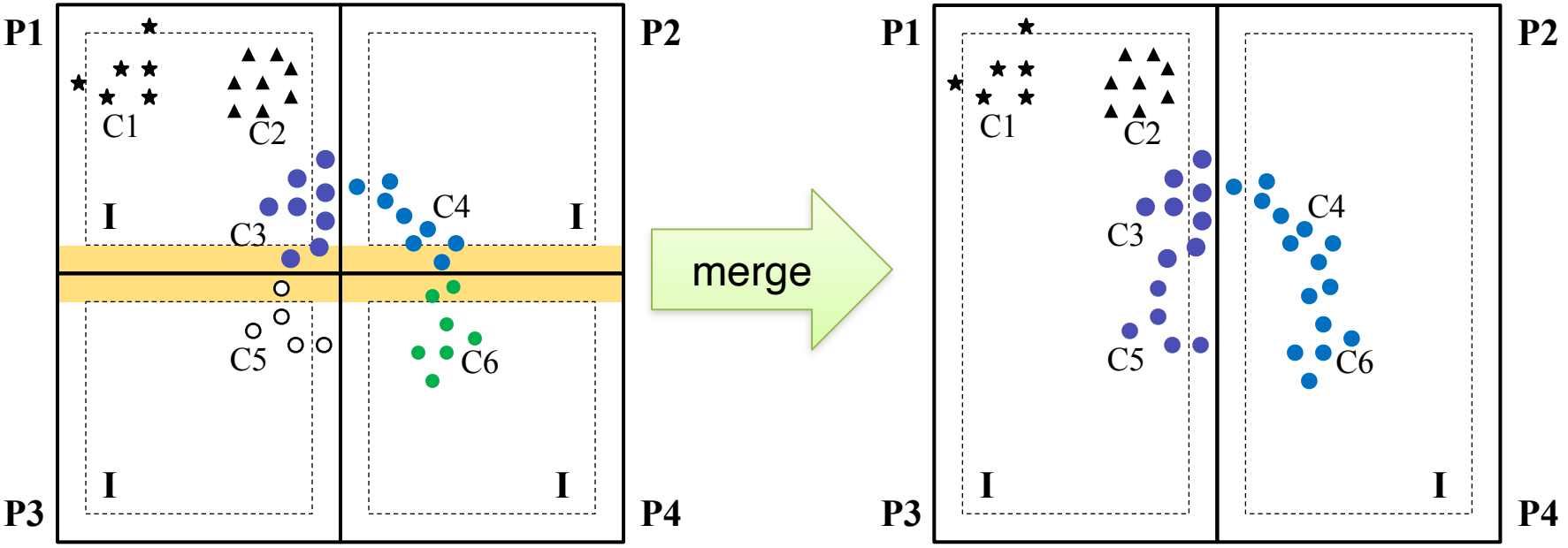
Roadmap

- Introduction
- Context: RDBMS, MapReduce, etc.
- New Extensions for Science
  - Recursive MapReduce
  - Skew Handling

Example: Friends of Friends



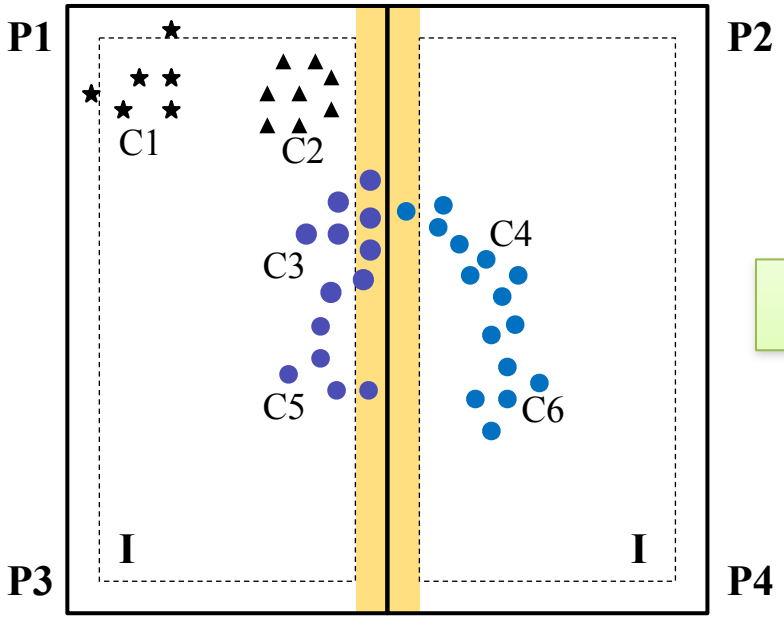
Example: Friends of Friends



Merge P1, P3
Merge P2, P4

C5 → C3
C6 → C4

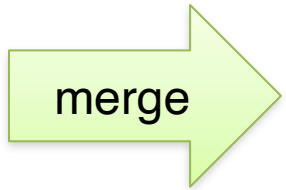
Example: Friends of Friends

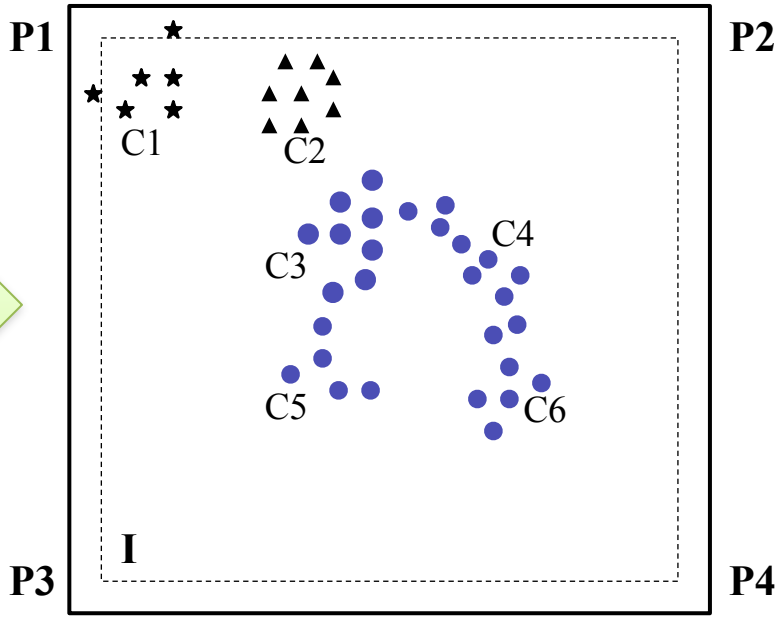


Merge P1-P3, P2-P4

C5 → C3

C6 → C4



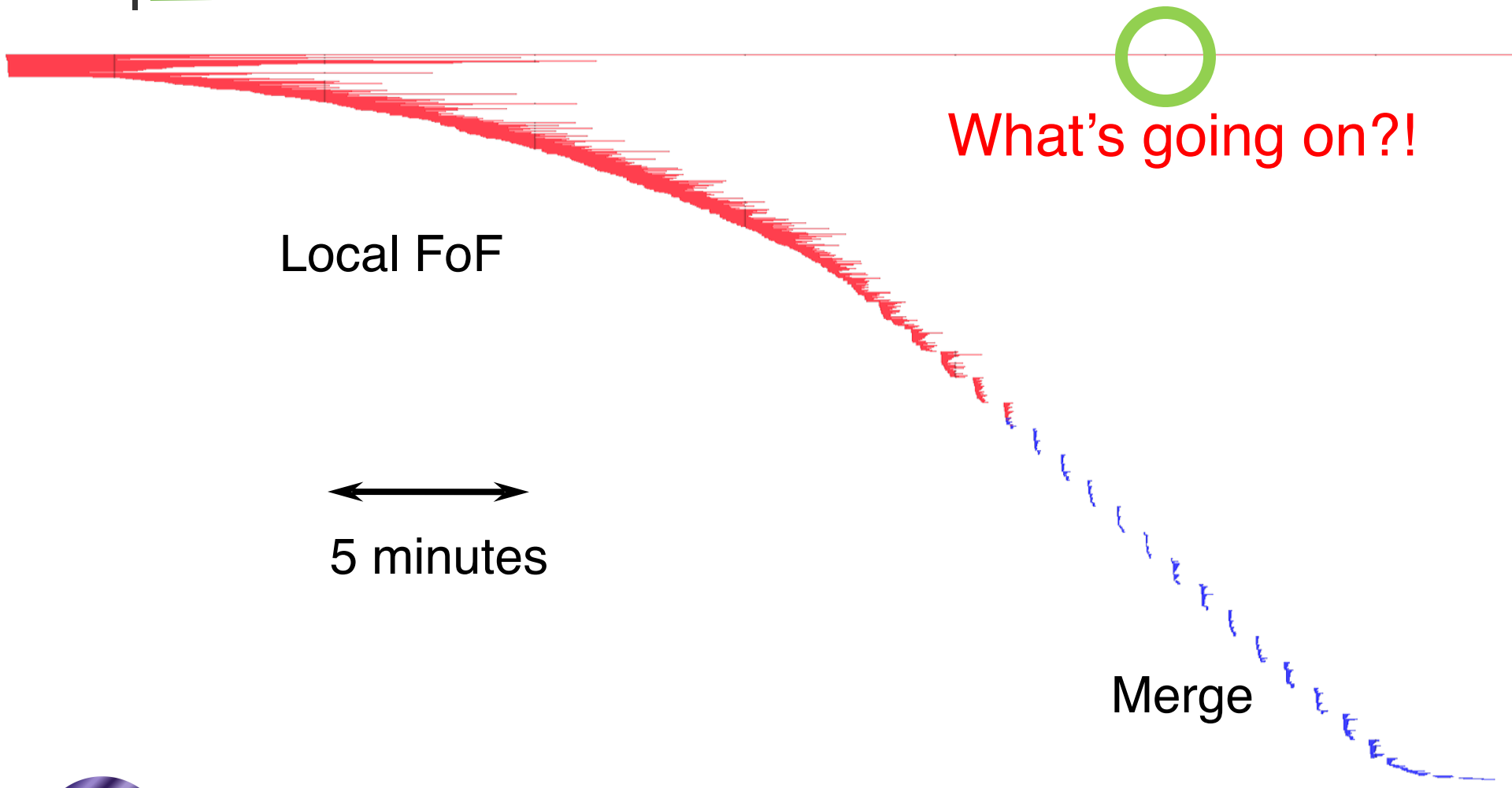


C4 → C3

C5 → C3

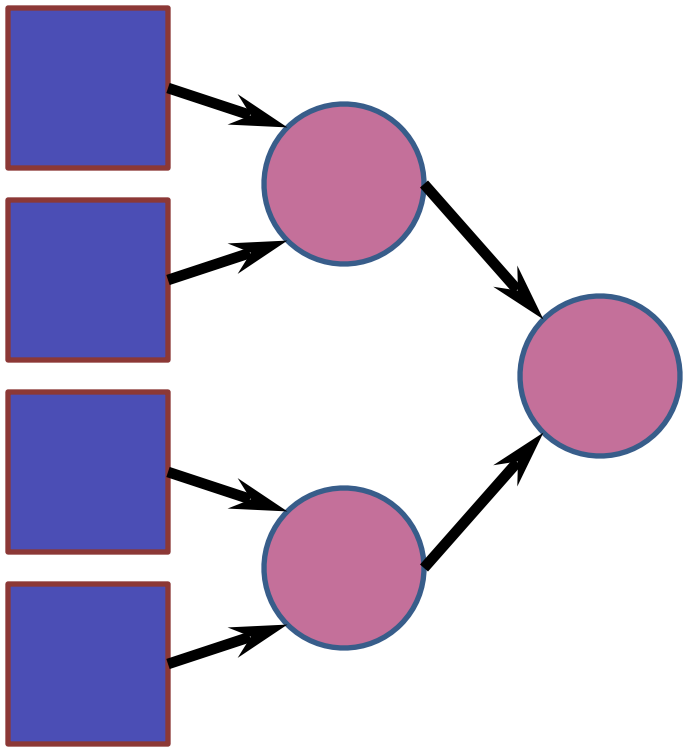
C6 → C3

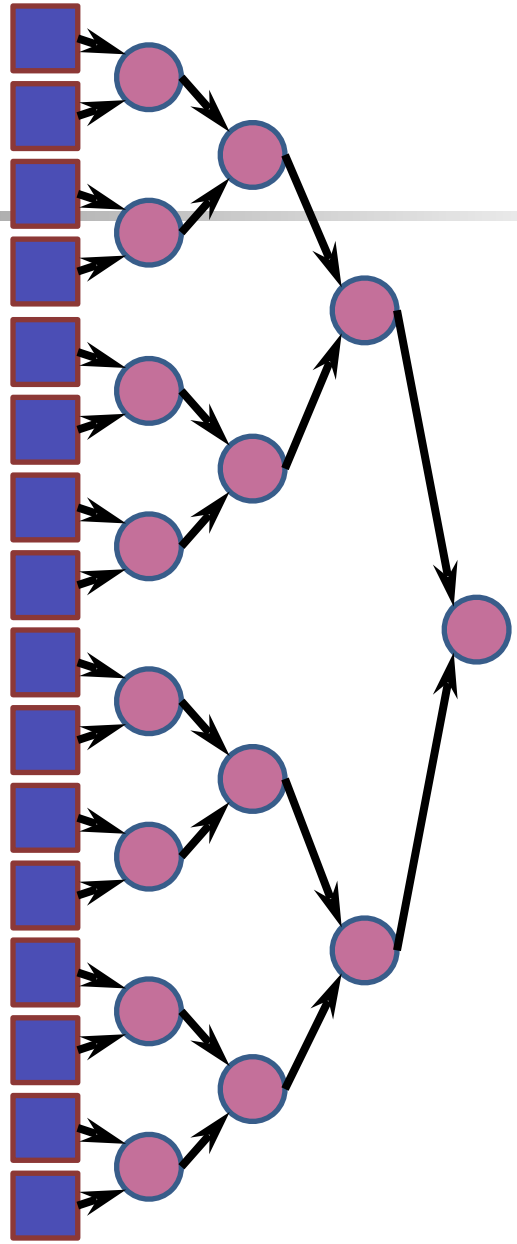
Example: Unbalanced Computation



The top red line runs for 1.5 hours

Which one is better?

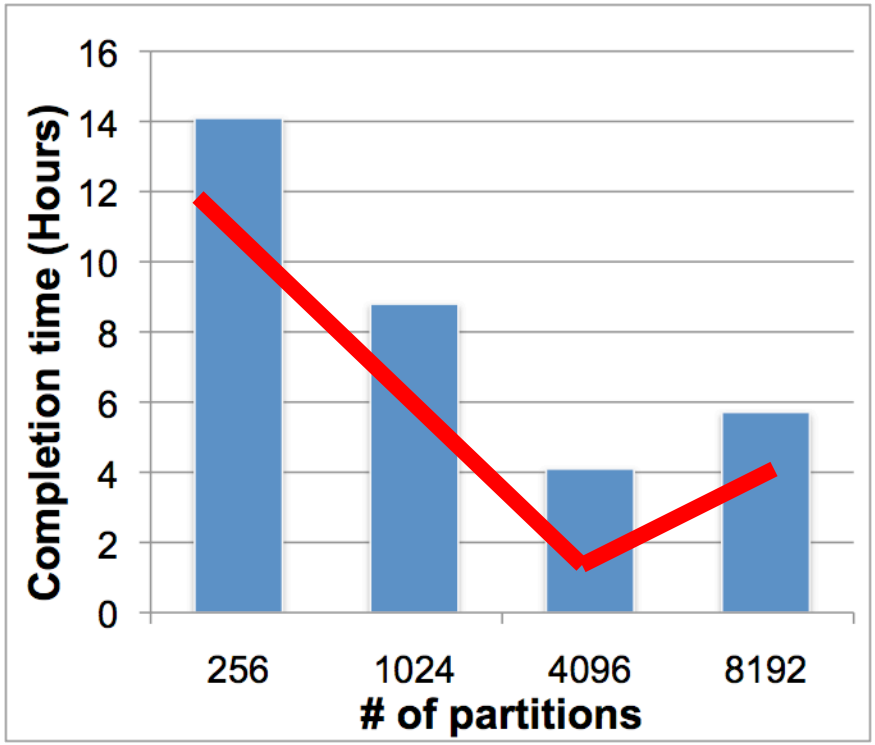




- How to decompose space?
- How to schedule?
- How to avoid memory overrun?

Optimal Partitioning Plan Non-Trivial

- Fine grained partitions
  - Less data = Less skew
  - Framework overhead dominates
- Finding optimal point is time consuming
- No guarantee of successful merge phase



Can we find a good partitioning plan without trial and error?

Skew Reduce Framework

- User provides three functions

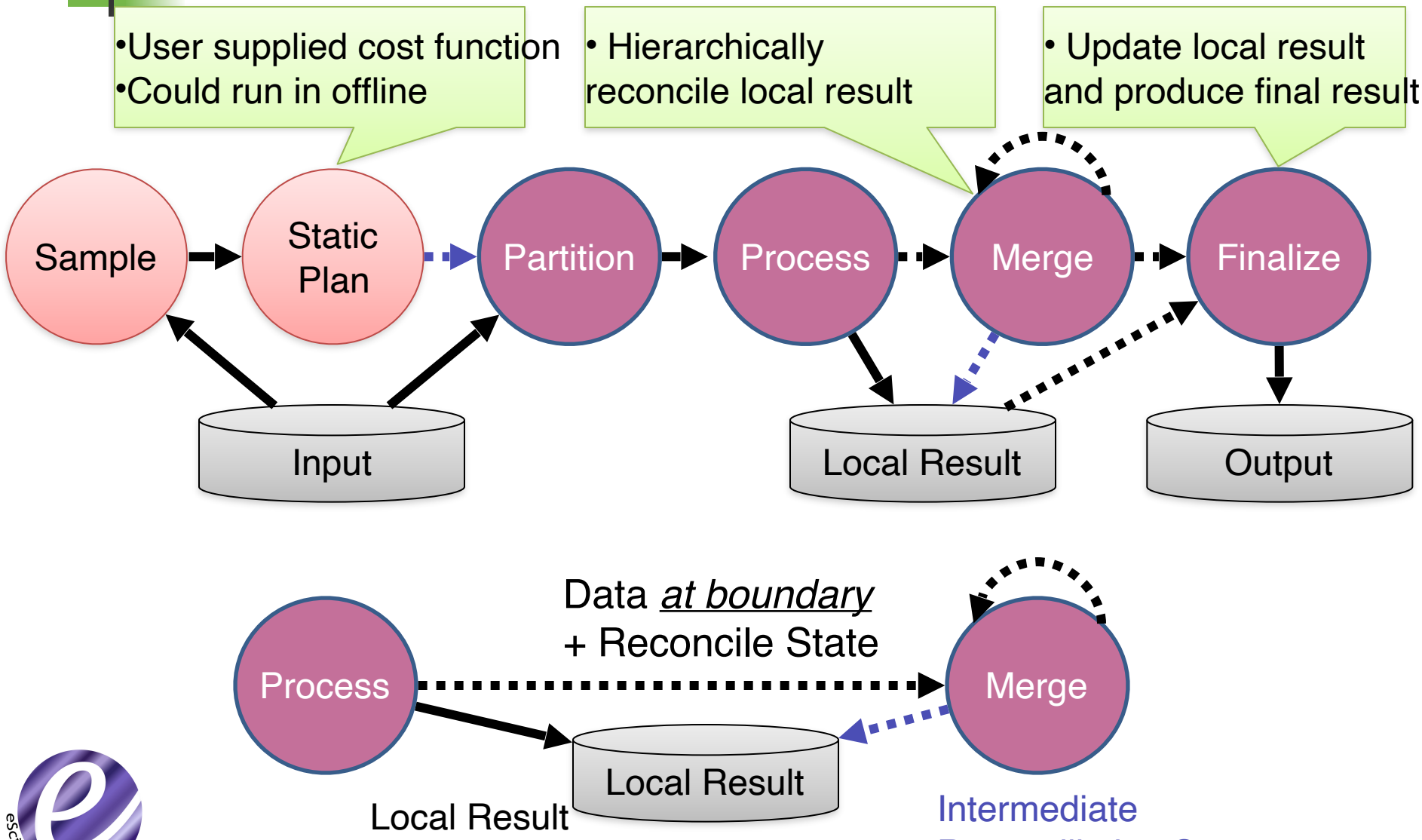
$$\begin{aligned} \text{process} &:: \langle \text{Seq. of } T \rangle \rightarrow \langle F, \text{ Seq. of } S \rangle \\ \text{merge} &:: \langle F, F \rangle \rightarrow \langle F, \text{ Seq. of } S \rangle \\ \text{finalize} &:: \langle F, S \rangle \rightarrow \langle \text{Seq. of } Z \rangle \end{aligned}$$

- Plus (optionally) two cost functions

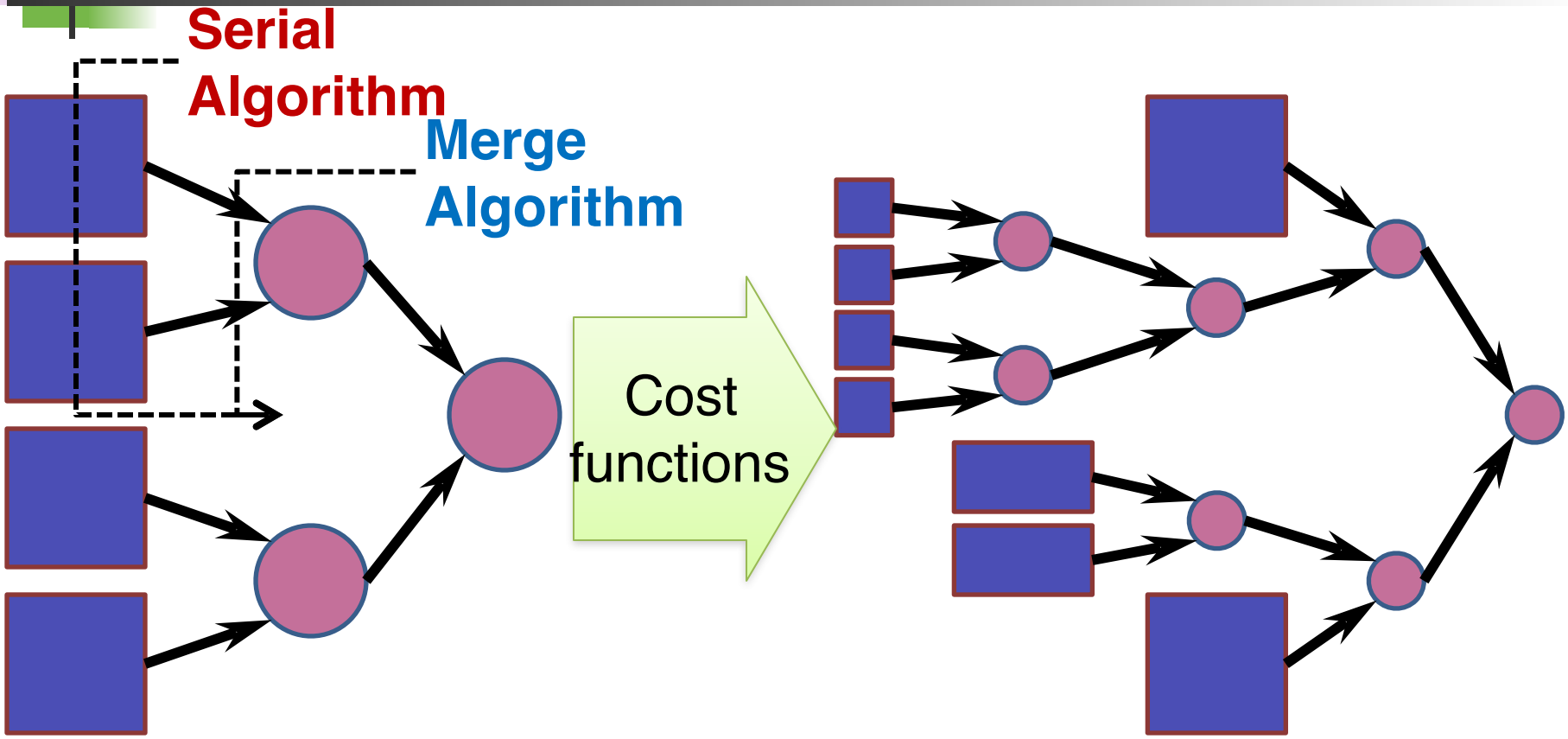
$$\begin{aligned} C_p &:: (S, \alpha, B) \rightarrow \mathbb{R} \\ C_m &:: (S, \alpha, B) \times (S, \alpha, B) \rightarrow \mathbb{R} \end{aligned}$$

S = sample of the input block; α and B are metadata about the block

Skew Reduce Framework

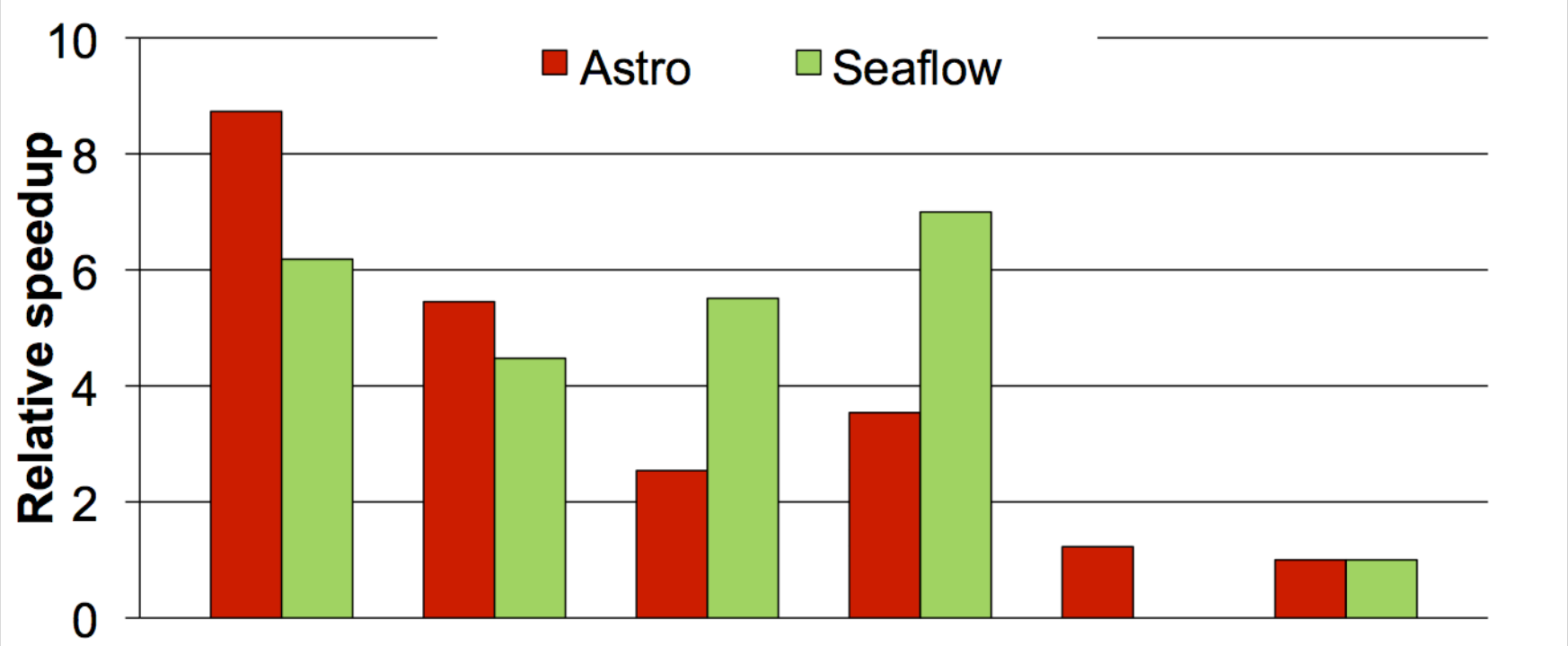


Contiribution: SkewReduce



- Two algorithms: Serial/Merge algorithm
- Two cost functions for each algorithm
- Find a good partition plan and schedule

Does SkewReduce work?



Coarse	Fine	Finer	Finest	Manual	Opt	
14.1	8.8	4.1	5.7	2.0	1.6	Hours
87.2	63.1	77.7	98.7	-	14.1	Minutes

Static plan yields 2 ~ 8 times faster running time

Data-Intensive Scalable Science

BACKUP SLIDES

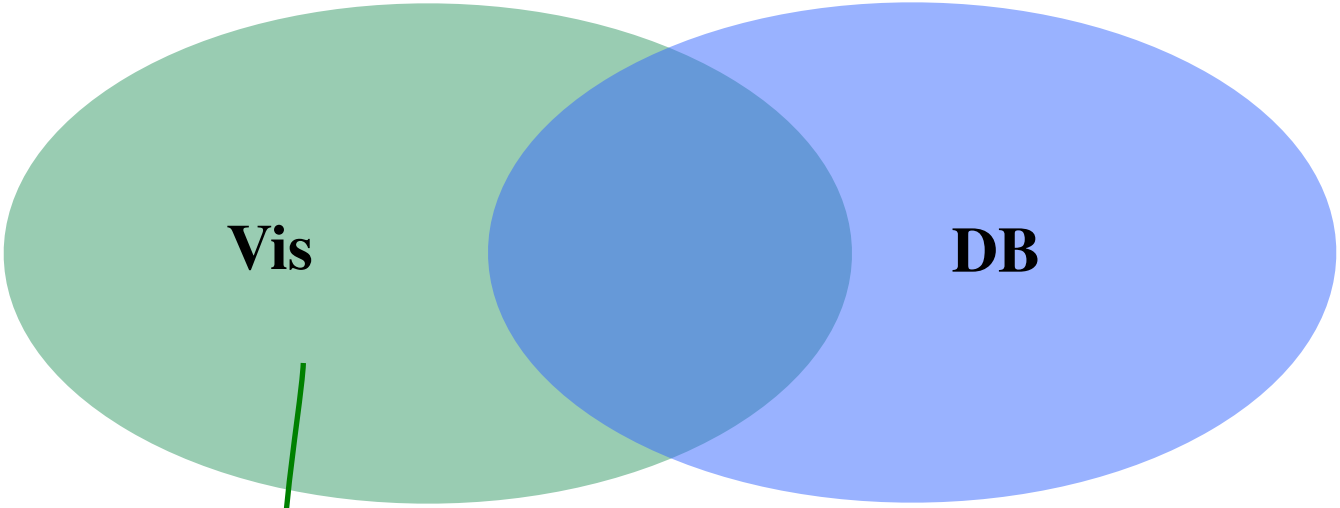
Visualization + Data Management

We can no longer afford two separate systems

"Transferring the whole data generated … to a storage device or a visualization machine could become a serious bottleneck, because I/O would take most of the … time. A more feasible approach is to **reduce and prepare the data in situ** for subsequent visualization and data analysis tasks."

-- SciDAC Review

Converging Requirements

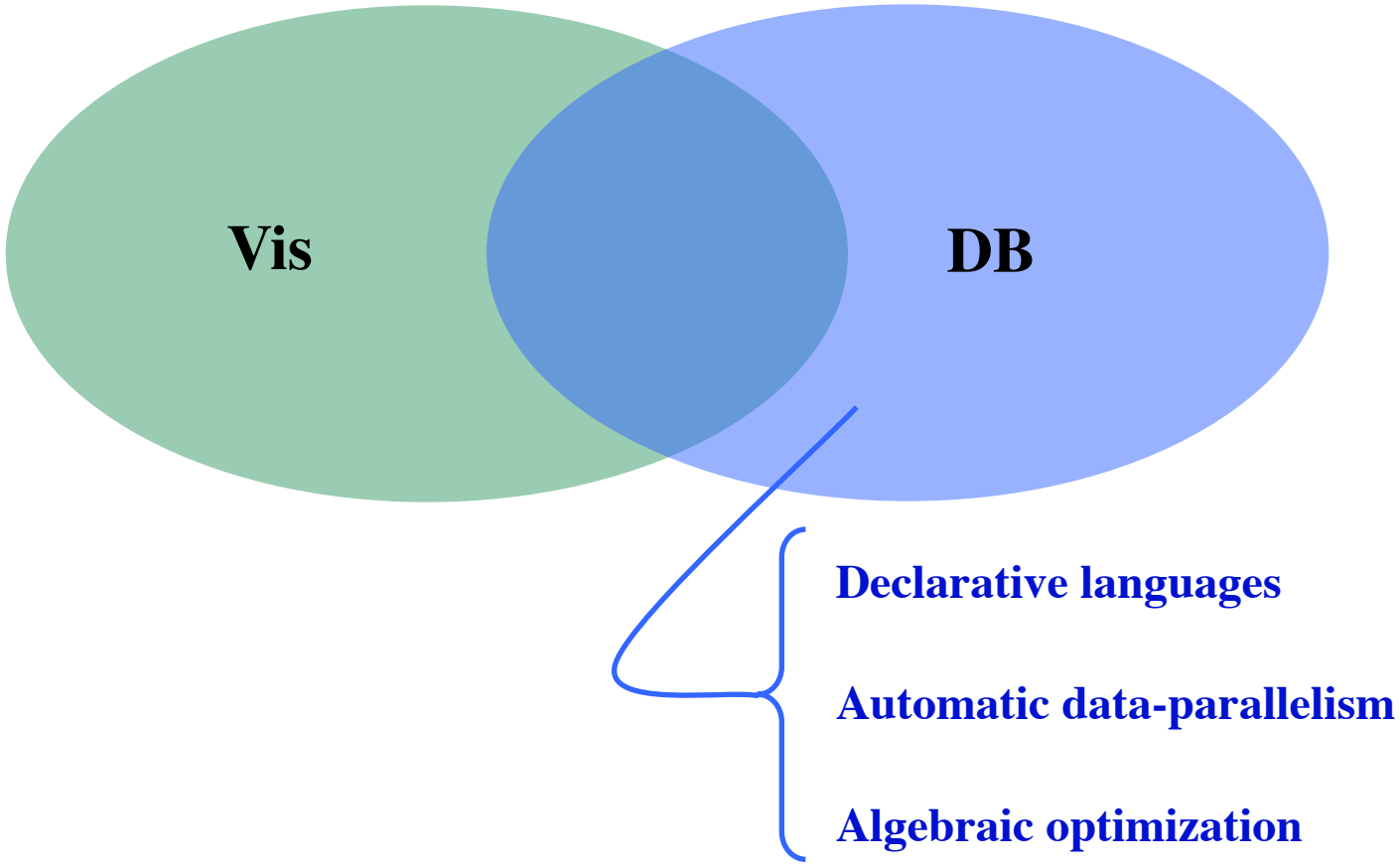


Core vis techniques (isosurfaces, volume rendering, …)

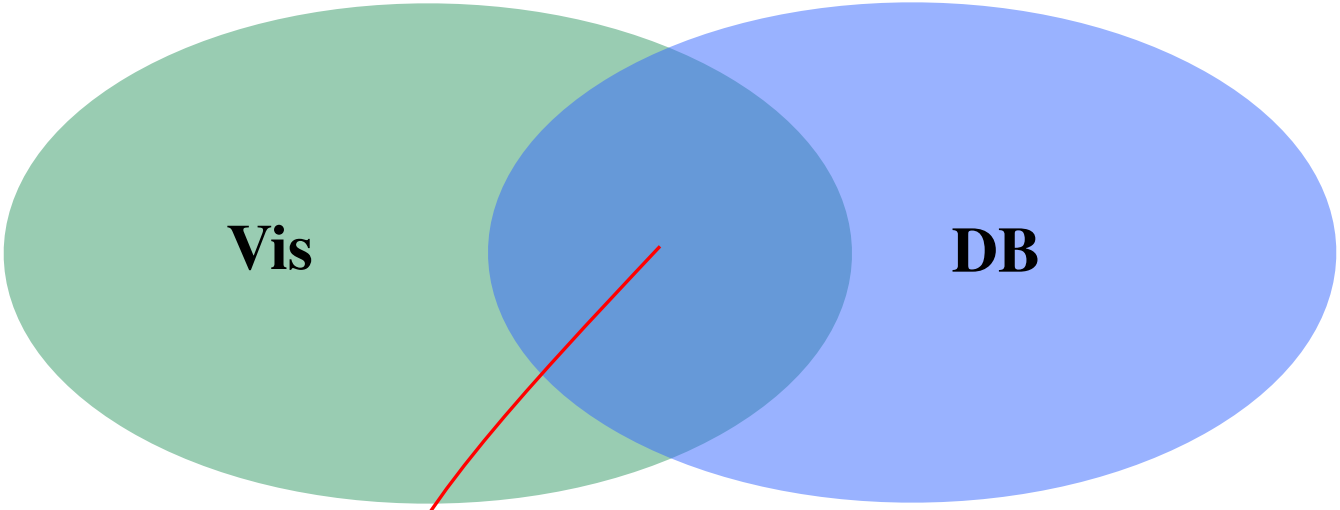
Emphasis on interactive performance

Mesh data as a first-class citizen

Converging Requirements



Converging Requirements



Vis: "Query-driven Visualization"

Vis: "In Situ Visualization"

Vis: "Remote Visualization"

DB: "Push the computation to the data"

Desiderata for a "VisDB"

- New Data Model
  - Structured and Unstructured Grids
- New Query Language
  - Scalable grid-aware operators
  - Native visualization primitives
- New indexing and optimization techniques
- "Smart" Query Results
  - Interactive Apps/Dashboards