Everything I know about data mining I learned from the ancient Egyptians

Or: How to spend money on a rock repository whose contents have mostly never been looked at again (and yet that contain some gold).

David Skillicorn

Queen's University

skill@cs.queensu.ca

Datacentric Grids

David Skillicorn

Queen's University

skill@cs.queensu.ca

An exemplar problem (at the small end):

Find the average value of galaxy brightness in the xray spectrum.

There are 100 gigagalaxies (new ones whenever Hubble is pointed in a new direction). Partially overlapped data about them is kept in ~30 big repositories.

Galaxies have about a kiloattribute: not all are recorded in each repository, and they tend to be scaled differently (e.g. to account for red shift, or not).

Some datasets can be downloaded; some have sql interfaces; some have home grown query interfaces.

The same object has different names (30+)

Today's solution:

Huge amount of figuring out dataset contents and properties up front.

Messy combination of downloading; generating queries; and postprocessing.

Poor solutions, and a lot of work to get any useful results (about 4 grad-student-months per result).

Tomorrow's Solution:

Pose a query (or write a simplish program), submit it to a datacentric grid, wait for result.

The datacentric grid is responsible for scheduling the required computations, validating the data accesses, resolving the partial results obtained from each site (hard problem!).

There are significant differences between datacentric grids and computational grids: naturally parallel/distributed/agent-based; more template driven; more reuse of computations.

Why not a data grid?

Large datasets start out distributed for reasons that do not depend on technology

Data is captured via different channels (web, store, phone)

There are legal, political, and social restrictions on data movement.

The cost per byte of storage systems imposes a maximum cost-effective size.

Large datasets are effectively <u>immovable.</u>

Data has a property analogous to inertia: easy to store, and easy to keep moving, but hard to change from one to the other.

Systems running data-intensive applications do not trivially have temporary space for large datasets.

Fetching remote data introduces long latencies at planetary distances.

So if datasets are distributed, and can't be collected into one place, how can we extract knowledge from them?

A <u>datacentric</u> approach to large data-intensive computations is needed – where code moves to data instead of data to code: a *datacentric* grid.

One of the big differences between datacentric grids and computational grids is that reuse is a much more important factor, e.g. same code, today's data; or slightly different code as the user converges on a justifiable hypothesis.

It's important to keep the knowledge extracted from a dataset, because it's likely to be wanted again (perhaps even by a different user).

Hence each (raw) dataset acquires a hierarchy of extracted knowledge – models, statistics, samples.
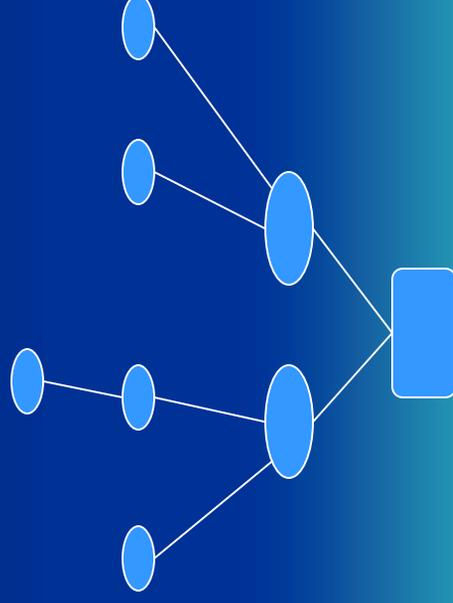
One of the interesting new problems is how to describe what can be found at a node of a datacentric grid.

A compute server can be described by a few simple parameters: # of processors, other available resources, and pricing.

A datacentric node, even holding one basic dataset, contains a number of other representations extracted from it. Even their formats may differ.

XML is probably not the solution.

- XML is too flexible so standards will not be adhered to (cf HTML)

- there's too much to say about repository contents

It may be necessary to send generic code to each data repository, and unpack it to an appropriate version based on local information.

The problems of high-performance computing are increasingly problems of data access and manipulation.

Distributed data mining is the canonical problem for the next decade.

We believe that computational grids are the easy case; that data access grids are a temporary mechanism because they do not scale; and that a more radical approach to data-intensive computation is needed.

www.cs.queensu.ca/home/skill/datacentric.html