Motivating Computational Grids

David Skillicorn

Queen’s University

skill@cs.queensu.ca

The big unasked questions about computational grids:

1. What motivates users of grid resources?

2. What motivates suppliers of grid resources?

3. What are the implications for the properties of grids?

4. What are the research questions?

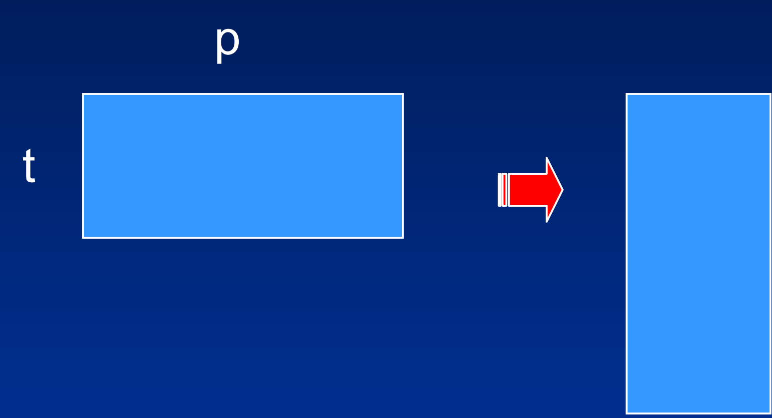
First, a simpler question:

Why use parallelism at all?

1. The task has a deadline – it needs to be done within a certain timeframe.

   -- hard deadlines, e.g. weather
   -- train of thought timing
   -- human time scales

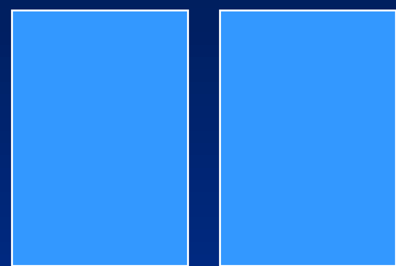
2. Brent's theorem – parallelism can be exchanged for time but not vice versa.



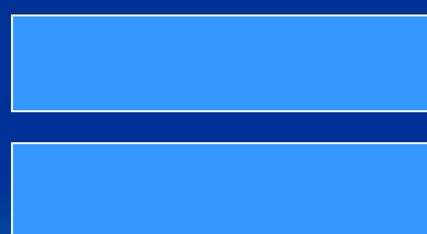
So it's always best to express computations in their maximally parallel form.

A set of concurrent cycles is much more valuable than the same number of cycles sequentially.

Therefore if two users, A and B, own cycles arranged like this:



they can both own something more valuable by agreeing to share like this:



BUT when A is using both, B gets no work done.

3. Parallel systems have more memory close to processors.

Memory latency is becoming the rate-limiting step in many computations. It makes sense to use a parallel system even for sequential computations that are memory-access bound.

What motivates users of grid resources?

A need for parallel cycles, but only occasionally.

Users with predictable requirements for parallelism, no matter how large, cannot do better than buying enough processors to handle their load.

Grid users must have peaks in their demand; the grid helps them to smooth these peaks.

Who has this kind of requirement?

NOT many scientific/engineering applications – which is why many users buy their own clusters or even single PCs. Performance is not as good, but time to completion may be better.

Computational grids often seem to have to be rebuilt for supercomputing conferences, which suggests that they are not in constant use in the interim.

What motivates suppliers of grid resources?

1. Generosity – e.g. Seti@home etc.
2. Share and share alike – I use your system to handle my demand peaks; in return I let you use my system to handle your demand peaks.
3. $$$ -- supplying grid resources is my business.
4. Our common organisation allows us to share our resources to smooth out our departmental peaks.

Generosity.

There can be a few programs such as Seti@home, but there can't be many – the unused PC computing time is a finite (though large) resource.

It's also quite wasteful – Seti repeats each calculation up to 6 times to protect against malicious miscomputation.

There are BIG security issues – are such systems really Carnivore@home? Is it plausible that they haven't been taken over?

Share and share alike.

Using our computing capacity to clip the peaks on loads makes a lot of sense – BUT only if we don't mind the other site seeing the data we consume and produce; the code we run; and when we run it and with what deadlines.

In some contexts this is fine. In others not so much – what if the site that executes my code is owned by my competitor?

Commercial supply of parallel computing cycles.

This is already happening, but disguised as web services – and usually free (because it often replaces some other, more expensive channel).

We're used to an economic model of computing costs with a capital cost up front, but each incremental cycle for free. Can we accept each incremental cycle costing real money? (This was the dominant model for the first 4 decades of computing.)

Sharing within an organisation.

This makes a lot of sense.

Multinational organisations tend to have resource in different time zones. This naturally creates demand ebb and flow, which can be shared globally.

And the security issues are much less critical.

In my view, these motivations suggest three different kinds of computational grids:

1. Free grids, in which the resources are donated. Major issue: security, requiring trust between strangers.
2. Public grids, in which the computations are in the public domain. Major issues: security; fairness in the pricing mechanism; existence of net suppliers of cycles.
3. Virtual private grids (VPGs), in which organisations share their loads internally. Major issues: determining priority.

What are the implications for the properties of grids?

<u>Security</u>: how can A execute a program on B's system so that:

1. A knows that B's system computes the right thing (no malicious hosts), and
2. B knows that A's program does not do anything destructive to B's system (no malicious users).
3. B doesn't learn anything from the fact that A is computing at this moment (no traffic analysis).

This is the biggest practical and research problem for computational grids.

Free grids only work because nothing bad has happened YET.

Public grids only work because the people involved have to know each other to get the technical side to work. (NB Globus assumes users may be malicious, but not systems.)

VPGs work because someone who violates security policies gets fired.

Pricing the use of resources.

In free grids this issue doesn't arise.

In public grids, the goal should be to be demonstrably fair – what's the exchange rate between my 200 slower processors and your 50 fast ones?

This probably doesn't require either the spot pricing or auction mechanisms that are being developed. Economic theory suggests rather that subscription based access is more likely.

In virtual private grids, there is still a need for pricing.

Currency is a way of asserting priority. If resources are priced in a way that reflects their effectiveness for organisational goals, then giving users budgets ranks the importance of their tasks.

The research issues here may be more subtle than those of public grids.

Summary:

An analysis of motivations for using and supplying grid resources suggests:

-- a small set of free grids,

-- a moderately sized set of public grids, but

-- the largest set of computational grids existing within single organisations.

used to manage varying demand for parallel computing cycles.

What are the big research questions?

Security: this is so intractable that it may, in the end, prevent the development of both free and public grids, except in simple forms.

Cost/pricing/accounting: the sorts of solutions needed seem to be quite different from those discussed in the literature so far.

Is eScience the right testbed to understand the grid?

Not really concerned about security;

Not really concerned about pricing (negative profit organisation);

Not really concerned about software engineering issues.

Hmmmm.

?

WERBUNG

For other papers about parallel models, parallel and distributed data mining, and datacentric grids, see

www.cs.queensu.ca/home/skill