Distributed Resource Allocation for Grid Computations

Peter Gradwell and Julian Padget

Department of Computer Science, University of Bath, Bath, UK
Market-based Resource Allocation

- e-Science scenario:
  - Physics Researcher doing Large Hadron Collision calculations
  - Requires: Software function; CPU; DataSet; Storage. Defined Budget & Timeframe
  - But... LHC Grid has 6000 Servers in 78 Countries

- Increasing take-up of the Grid suggests emergence of e-Social Science, e-Health, e-Engineering, even e-Music

- Standard solution (for optimality) is the Combinatorial Auction (CA)
Combinatorial Auctions

- In complexity terms they are NP-Hard
- Current limits are (Sandholm): “tens of items and hundreds of bids per min”
- Small improvements keep on coming (Sandholm, Parkes), or can clear in polynomial time with a bound of the optimal solution (Jennings+Hu(?))
- CA requires complete control – a single auction space
- Assertion: CAs are difficult to apply to resource allocation on large disparate grids:
  - Bundling problem is too large to solve
  - Grid nodes and bidders are distributed – a single combinatorial auction seems impractical
Distributed Auctions

- A market-based solution: a Grid Commodities Market (GCM)
- Distributed auctions enable cross-fertilisation of a wide range of traders and buyers – as found on the Grid.
- Intelligent (middle) agents assemble bundles against customer requirements (actual or prospective)
- Trader agents are profit motivated.
- Traders may not sell all their bundles – so there is natural wastage in the system.
- GCM is suitable for open grids as no relationship is required between trading parties
Traders perform bundling, but many of them, so might distribution cause time to approximate linear?

System may not be Pareto-optimal, but it should construct useful bundles.
How to compare?

- CA is an algorithm
- GCM is a complex system
- Analytical approach unrealistic
- Build a model? Have to do that anyway
- Simulate:
  - Collect empirical evidence
  - Use standard test cases (CATS/Stanford)
- Second approach: make CA faster but non-optimal:
  - Explore sensitivity of optimality to allocation
  - Cache allocations
  - Return previous similar allocations subject to proximity bound and analytic continuity
  - At what point, if ever, will quality of allocations cross?
What is close enough to optimal?

- Currently: investigating proximity of a bundle to the (strongly) Pareto-optimal bundle.
  - CA performance is highly dependent on the heuristics used in the computation (CABOB: Combinatorial Auction Branch On Bids).
  - The GCM approach may not produce a Pareto-optimal solution since it has incomplete information.
  - Can we use heuristics to improve GCM?
- Can GCM traders remember popular bundles and assemble them pre-emptively? Is market memory better than zero-intelligence?
- How does re-sale/re-circulation of items impact market dynamics?
- When is a middle agent bankrupt? How to reallocate rights to resources that dead traders have bundled?