

Grid is not THE answer *but it is the central piece in the puzzle*

Why not?

If your IT infrastructure suffers from low utilisation, or you aren't able to hit SLA's for your overnight batch runs then you may benefit from implementing a grid based solution. However, Grid is not a silver bullet, not by itself at least. It is important to realise that grid is only part of the solution, and can - must - become a core part of a technology strategy along with Data caching, HPC technology and (gulp) organisational change.

Grid is a great enabler. It was designed with this in mind. Heterogeneous, service oriented, ubiquitous - grid has taken the best of HPC and distributed computing to build a pervasive middleware that enables hardware virtualisation. Think of it as an operating system that has the ability to take in work, schedule it, distribute it to any appropriate resource and to guarantee that it will complete (if all of the parameters are set correctly) once submitted. This virtualisation offers some fantastic capabilities.

For the right kind of work grid provides a near linear scaling capability. Any job that can be split into an independent set of tasks can be scaled up to the maximum number of tasks. But there is of course a level of task granularity that doesn't warrant distribution. Usually this would be when the task run time is less than the task distribution time, and when your task needs a lot of data this can occur more quickly than you think.

For the wrong kind of work grid makes no difference, and can even make things worse. It makes little sense to distribute a single job with no parallel parts to a grid (unless of course your key requirement is resilience) unless one of your grid resources can run it quicker than your local machine; particularly if your calculation requires a lot of data. In this case a computation may be waiting for database access before it can do any work. While commercial grid offerings give some assistance in this they freely admit that data is not their core business. A tactical grid-only solution may improve performance for a while, but in reality all it does is move the bottleneck to the data, and scale limits will be eventually reached again.

Grid can drive up utilisation through intelligent scheduling and resource allocation, and if you can encourage users to share grids then it can make more resource available to them at different priorities at different times, thus helping IT (by making machines more busy) and the business (by getting more things done).

If you have performance problems, they can certainly be addressed by a grid, and as a tactical solution this may be a good start. However fundamental performance problems should be tackled at source as part of any remediation. All this is why grid is not THE answer.

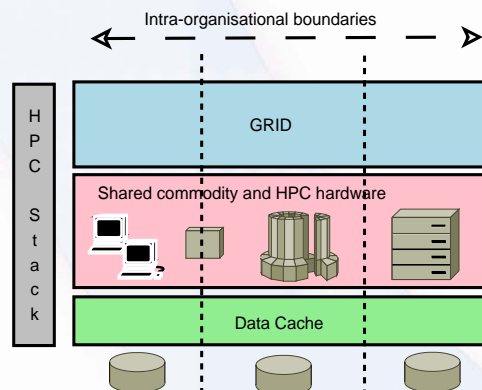
Intelligent glue

Data movement problems can slow a grid down, and sharing seems like a good idea, but is difficult to achieve. Grid may not be THE answer, but the answer does include, in fact it rests on, grid. THE answer is a strategic application and organisational architecture that as a sum of parts is greater than the whole; for simplicity we will call this the HPC stack.

Lets address each problem one at a time:

- **Making best use of available resources** - grid middleware makes intelligent scheduling, prioritisation and resource allocation available right “out of the box”.
- **Getting data to the right place at the right time** - Data caching solutions offer an interface that separates the database from the compute nodes, improves locality of reference of data and compute and uses agent technology to reduce the data bottleneck.
- **Sharing resource** - Everyone likes to borrow, but not many like to lend. Grid software makes lending and borrowing resource easy. The difficult part is overcoming the organisational barriers to sharing. Part of this is the process of creating buy-in by engaging business users in the re-engineering process and understanding their needs, thereby addressing their requirements. Part of it may come from on high.
- **Not enough compute** - To some extent, grid addresses this by making more resource available. But grid is not high performance computing, it is high throughput computing. Real compute can be made available through specialist technologies such as programmable chips, multi-core hardware and supercomputers. Suddenly, grid makes all of this possible by providing a virtualisation layer that hides this resource away, and an intelligent resource allocation layer that knows where certain jobs can be serviced. Those jobs that it made no sense to distribute - well now it does, because they can be distributed to a single grid engine that just happens to have a multi-core chip underneath it!!

The glue that holds all of this together is the grid platform. A high throughput platform that is heterogeneous, always on, resilient and adaptive. Without the intelligent glue that is the grid, none of this would be possible.



In summary: a tactical grid implementation will solve an immediate problem and seed the solution; a strategic grid implementation will enable it; and a full HPC stack which includes Grid, HPC technology, commodity hardware and caching will ensure that your architecture can scale for the future.