Grid Performance with tkeden

University ID: 0112806

Abstract

Grids are broadcast as the framework to perform rapid data analysis. This is often required to gain a competitive advantage in industry. The speed of any such network, however, depends largely on its architecture. In this paper, I describe a model that allows users to explore the effects on a Grid's performance of altering its arrangement. I start by discussing the relevant subject area, followed by a critique of empirical principles and the tkeden toolkit in modelling this state.

1. Introduction

Many situations require quick and accurate analysis of ever rising data volumes (Schmerken, 2005). Grid computing is seen as the de facto standard in meeting such requirements. Already in widespread use, figures indicate investment in this technology will reach $4 billion per year by 2008 (Harris, 2004).

The extensive use of grid computing has uncovered several challenges. This includes workload management. Specifically, the overhead of moving data between the Grid's resources means shorter processing times are not guaranteed by supplying more hardware to the network, i.e. performance is not scalable. Therefore, grid implementations often involve a tradeoff between the capital employed and performance.

This paper presents a model that considers the standard, three-tier Grid architecture. The tkeden toolkit was used for this project, which proved helpful to the assignment. It afforded an open development approach to modelling, which is useful when modelling an area with inadequate theory such as grid computing (Rungrattanaubol, 2002).

The model is intended to educate the user on the workings of a simple Grid system and therefore, assist any decisions to be made during an actual implementation. This will reduce associated costs and give a clearer understanding of likely performance.

2. Grid Computing

Grid computing enables the virtualization of distributed computing and data resources such as processing capacity to create a single system image, granting users and applications seamless access to vast IT capabilities (Yoo, 2004).

Proprietary grid infrastructures typically run on dedicated clusters (Lees, 2005). However, grid computing across a disparate and varied set of machines is more complex. The geographical distribution of machines, for example, exacerbates the already serious problem of data distribution (Harris, 2004). Furthermore, a broad variety of performance profiles (in terms of computation and communication) makes optimal scheduling more complex. Additionally, for many implementations, one must consider any residue left on host machines; in any highly-regulated industry, this is of particular concern from a security standpoint.

The inner-workings of a Grid are influenced by its purpose. Whether a proprietary or distributed infrastructure, Grid networks are classed under two categories: *data-intensive* or *process-intensive*. Data-intensive Grids are typical of those found in the bioinformatics field, while process-intensive Grids are typical of the financial services industry.

3. Modelling Study

3.1 Description

The model takes the shape of a standard, three-tier Grid architecture (see Fig. 1). The tasks entered by the user are independent of one another. This draws a parallel to the typical use of Grid technology for Monte Carlo simulations in the Financial Services industry.

A workload is fed into the Grid via the single Broker (top tier), which forwards individual work packages to a Coordinator (middle tier) depending on which is the least-loaded at the time. These are responsible for delegating the single tasks to the Calculator(s) (bottom tier) for execution. Again, the Calculator is selected on a least-loaded basis.

This model judges the performance of a Grid by considering data transfer and resource latency. To remain loyal to the real-world, the model also simulates the process of returning a "result" to the Broker. While no actual solution is provided, an overheard (in time) is incurred. Overheads are also incurred when the user removes resources that hold tasks.

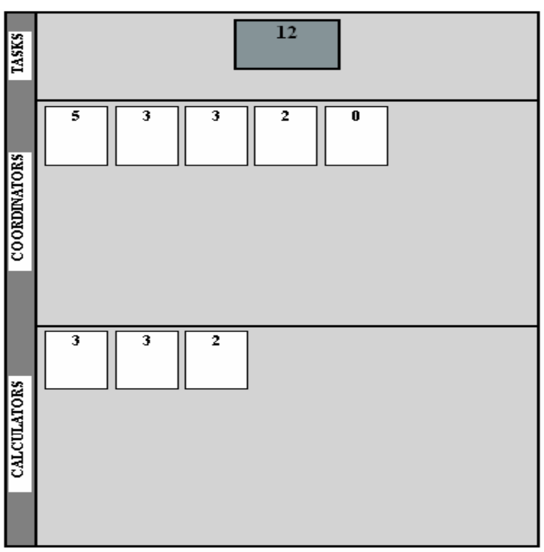

Figure 1: Grid layout.

A more detailed description is contained in the Documents folder of the model. This includes a walkthrough of the Grid while in operation.

3.2 Model Applications

The model embodies what is directly experienced, at an administration level, when working with Grids. This is the movement of individual work packages. The intricacies of work scheduling and processing etc. are usually hidden from any user. Therefore, this model is suitable for use in any situation where the state of an entire Grid network is of importance, instead of the actual processing involved.

In building this model, two such situations were considered:

- Education
- Business

Users may track workflow through the Grid architecture, which is aided by the way the model prints messages about its operation (see Fig. 2). This is useful for those who are new to Grid computing practices.

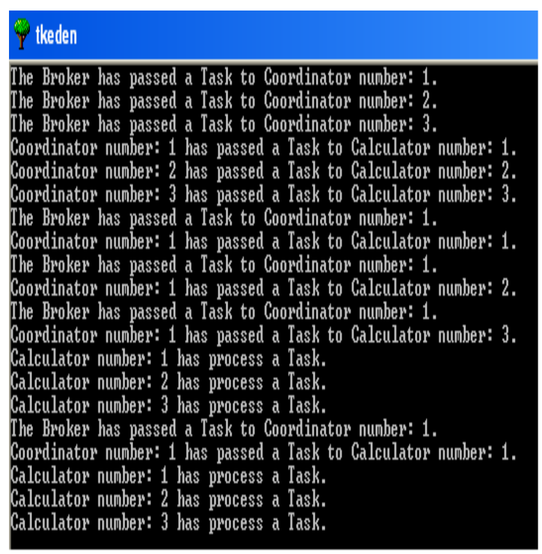

Figure 2: Messaging Output.

The findings from this model also have a commercial impact. As was an initial intention of pursuing this model, a more well-informed decision can be made as to the amount of resources to employ in a Grid to process an anticipated workload. The model can be reused using various amounts of Coordinators and Calculators to find the best combination for an identical workload.

As is a beauty of Empirical Modelling and the tkeden toolkit, the user is able to explore what happens when you make changes while the model is running. For example, the user can bypass the Broker by manually adjusting the loads of individual Coordinators/Calculators. This would not be allowed in traditional models. In fact, users can change the model's rules altogether. At the moment, the amount of Coordinators and Calculators is capped at 21. This can be changed by tweaking the code in the *toolbar* file.

3.3 Related Models

In pursuing a further understanding of grid computing through working with tkeden models, I advise referring to Declan O'Gorman's model: *Grid Computing – An empirical perspective*. The model offers an alternative insight into Grids by focussing on a different flavour of Grid. The node arrangement is fixed and the model's focus is not the time taken to complete a particular workload. However, it does give the observer an appreciation of the possible scope of Grids

Several features of my model's code, namely the use of *execute* and *display* can be further studied in Karl King's: *drawSlide* and Simon Yung's: *Room Viewer* models.

The distinctive trait of this model over other Grid models is how extra agents can be created by the user. In doing so, there is a closer consideration for *drawSlide* in this model.

3.4 Model Limitations

This model assumes tasks are identical and therefore, have an equal "processing duration" set by the user. This is not usually the case and its accommodation will involve considerable further development. Any autonomic behaviour and work scheduling, which are features of some Grids, are also beyond the current scope of this model.

In addition to those features deemed beyond the current scope, I have capped the numbers of Coordinators and Calculators at 21 each. This was made purely for visual purposes and can be adjusted to allow the modelling of larger Grid systems.

4. Empirical Modelling & tkeden

Empirical modelling offers a framework for computer-based modelling based on three key concepts: *observation, agency* and *dependency* (Rungrattanaubol, 2002). This fits well with Grid Computing practices. The *Agents* of this model are the Broker, the Coordinator(s) and the Calculator(s). The *Observables* include: the number of Coordinators, Calculators, tasks, various speed parameters and the overhead of data transfer. *Dependency* is illustrated in the visual layout of the Grid. This is loyal to the manner in which Grids are drawn, conceptually, as a series of identical resources distributed in a consistent arrangement.

Open-development modelling is also afforded by Empirical Modelling principles, which was useful as I found no adequate theory to grid computing. This field, after all, is vast and grids come in various forms to suit its application as previously mentioned.

4.1 The tkeden Toolkit

The tkeden toolkit proved suitable for this model because of its interactive nature, which it is hoped will also aid the learning process of Grid computing. Users construct a Grid by adjusting several settings to reflect real-world conditions (see Fig. 3). This makes the modeller a designer of the artifact as well as the user.

The empirical principle of dependency is also well-implemented in the tkeden toolkit. This feature allows the model to maintain the physical layout of the Grid's resources as is usually construed by an external observer to a Grid, i.e. as a uniform arrangement.

Figure 3: Settings toolbar.

The *EDEN clocks* feature of tkeden allowed animation as part of the model, which is central to the observing agent's capacity to study the Grid in a continuous state.

4.2 Critique

An Empirical Modelling approach to modelling can be misleading if taken in isolation. Trust in empirical evidence does not give sureness because experience can be deceptive and ignore certain use cases. In the case of this model, a specific kind of Grid is considered in a controllable level of detail and therefore, is adequately true to reality. More complex Grid models, however, may not be so faithful to their real-world counterparts.

One other key feature of Empirical Modelling is its idea of *openness* in terms of interaction with the model. To this end, the model provides a toolbar to adjust key parameters. To explore certain algorithms effectively, however, user intervention ought to be confined. In this model, such algorithms are those that find the least-loaded Coordinators and Calculators.

The nature of this model meant the dynamic creation of agent windows using the *execute* command. This entailed intricate coding. Therefore, tkeden was deemed inappropriate for modelling this particular feature of the Grid. However, this issue can be avoided by modelling Grids that do not allow such flexibility as in Declan O'Gorman's model: *Grid Computing – An empirical perspective*.

On a positive note, the tkeden toolkit does allow on-the-fly development that is beneficial in prototyping new features. Future versions of this tool should, however, remedy the need for such intricate coding when creating multiple windows dynamically.

5. Conclusion

A model has been presented that permits users to explore a typical Grid infrastructure and study the effects on its performance of altering its setup. This meets the objectives it was built to satisfy; however, there is much room for extension.

5.1 Extensions

There is scope to enlarge this model in several directions to include permitting:

- A variety of tasks,
- Interdependencies between the actual tasks, and
- Autonomic behaviour in the Grid.

Further study of this model's code clarifies the fact it was built in an incremental fashion. This approach can easily be sustained to implement the extensions above.

Acknowledgements

I would like to acknowledge the priceless advice, encouragement and insight received from Meurig Beynon, Steve Russ and Antony Harfield throughout this project.

References

Harris, D. (2004), Grid: A Work in Progress with $4bn Potential [online], Available from: http://www.gridtoday.com/04/0524/103263.html

Lees, W. (2005), The Evolution Of Grid Computing At UBS Investment Bank, Lecture presented for Current Uses of Computers module at the University of Warwick, Coventry.

Rungrattanaubol, J (2002), A treatise on Modelling with definitive scripts.

Schmerken, I. (2005), Data Volume Skyrockets [online], Available from: http://www.wallstreetandtech.com/showArticle.jhtml?articleID=164903661

Yoo, I. (2004), Grid & Utility Computing in the Financial Services Domain [online], Available from: http://www.caip.rutgers.edu/news/bios/ikyoo.htm