

A Solution to the Information Retrieval Problem in Desktop Grid Task Assignment Using the Force Field Model

Edscott Wilson García¹ * and Guillermo Morales-Luna²

¹ Instituto Mexicano del Petróleo, Programa de Matematicas Aplicadas y Computacion, Av. Lazaro Cardenas 152, 07700 Mexico D.F., Mexico. Tel: +52-55-9175-8216 Fax: +52-55-9175-7458

`edscott@imp.mx`

² CINVESTAV-IPN, Computer Science, Av. Instituto Politecnico Nacional 2508, 07360 Mexico D.F., Mexico. Tel: +52-55-5061-3759 Fax: +52-55-5061-3757

`gmorales@cs.cinvestav.mx`

Abstract. With pondered task assignment methods which utilize the dynamic state of the grid resources to solve task-to-processor mappings, an important problem arises from the difficulty of collecting the information from the remote nodes. In this paper we show that with the use of the force field method for task assignment, the information collection is not a problem and the method outperforms random non-repetitive task-processor mappings under desktop grid conditions. The solution is presented in both analytic and experimental manner. The simulation results illustrate the effectiveness of the force field method.

keywords: desktop-grid, grid-simulator, force-field, task-assignment

1 Introduction

The motivating scenario for the work presented in this paper are the large desktop grid environments which are being integrated and growing every day. In the implementation of task assignment strategies which take into consideration the network and CPU load distribution throughout the grid, one of the most difficult hurdles to overcome is the process of collecting information which characterizes the system as a whole.

The challenges of executing high performance computing on a virtual parallel computer composed of desktop PC's is a very different scenario than that of government funded scientific laboratories or private universities pooling their computation resources into large scale grids, as reflected in works such as DI-Gruber [1] or the agent approach detailed in [2]. To make the distinction clear of the differences between these approaches and the work presented in this paper, the distinction pertinent to a *desktop* grid is clarified below.

* Supported by the Basic Science Competence of the Instituto Mexicano del Petroleo

Current technology trends indicate that desktop grids for HPC will become applicable in the near future: just as the Internet began as an exclusive arena for universities and government supported research institutions and has now become mainstream, the tendencies in HPC will follow suite and become routine in the research and development areas of private corporations. How to take advantage of this desktop infrastructure for the resolution of complex scientific applications with increased demand for computing power and growing datasets is an important consideration for the future. Currently HPC capability has been spreading in such manner that the average commodity-off-the-shelf personal computer now surpasses the multi-million dollar HPC computer of not so many years ago. This, in turn, moves the focus of HPC from plain *number-crunching* to more elaborate data intensive applications in parallel environments. Recent developments such as GridDaenFS [3] reflect the interest in solving issues regarding data intensive applications. Furthermore, task assignment strategies such as the Force-Field Method [4] present novel methods to take into consideration the number-crunching ability of the individual nodes within the computer grid as well as communication cost.

Looking beyond blind strategies such as those analyzed in [5], one of the main issues with weighted task assignment methods is the problem of collection information: this is necessary to characterize the state of the communication channels and remote CPU availability. Task assignment should minimize the wall-clock time for the job. The time taken to gather the information plus the execution and communication time for the job should be less than that produced by blind strategies in order for the pondered method to be of any practical use. This is the main problem addressed in this paper, using the force field approach detailed in [4].

We introduce the theoretical background for a force field simplification algorithm and statistical results which demonstrate our claims. Our focus is on measuring the effectiveness and performance of such a framework, as well as gaining insights about the behavior of the methodology as the desktop grid size grows to systems with as many as 50000 nodes.

- We describe the considerations which need to be done to solve the information retrieval problem in desktop grids.
- We evaluate the proposed considerations by means of a simulator of the empirical conditions which characterize a desktop grid.
- We demonstrate that with the proposed considerations, the force field approach is a feasible and effective method for task assignment methods within desktop grids.

The rest of the article is organized as follows. We first provide background details of the problem that we address. We then discuss the tools used develop the force field simplification methodology and to perform the experiments, as well as the model used for the parallel computer to be emulated. Section 3 deals with the conditions of a desktop grid and the considerations which allow the construction of a simplified force field model. Section 4 contains the description of the experiments, the results we achieved, and the comparison of blind methods versus the pondered force field strategy. The rest of the paper focuses on our conclusions.

2 Background Information

As pointed out by Balakrishnan *et.al.* [6], while P2P networks are autonomous and scalable, they lack the required understanding, coordination, and scheduling capabilities to support advanced applications. With this in mind, Zhuge [7] points out that intelligent services rely on the ability to cluster and re-cluster heterogeneous resources and thus in the future interconnection environment resources must flow from high to low energy nodes: this automatically requires appropriate on-demand logistics because the energy difference reflects a need for flow. Simulating a desktop grid as an biological organism may be too complex at this point; instead the grid can be simulated as a universe with gravitational forces determining a force field. This force field determines where task assignment takes place. Recent theoretical advances in physics

hypothesize the existence of particles known as gravitons, which travel between masses to create the force of gravity. For the force field task assignment, gravitons would be the analogy for the information packet from the remote node to the computer doing the assignment. Just in the same manner that reception of gravitons from the furthest corner of the universe is not a problem for the motion of celestial bodies, reception of information packets from the furthest corner of the desktop grid should not be a problem for task assignment which proceeds under the force field model.

2.1 Desktop Grid

When the resources of a set of computer clusters or shared memory multiprocessors —dedicated to the task of scientific computing—, are pooled together to conform a computational grid, the result is a relatively static configuration where the interconnection network characteristics and the individual computational resources are previously known. A *desktop grid*, on the other hand, is the result of considering a large collection of personal computers where the principal utilization of the processors is not scientific computing but everyday tasks such as word processing, e-mail or Internet browsing. In a desktop grid, processor can appear or disappear from the grid without notice, and the individual characteristics of the available computing power and communications cost change in an unpredictable manner.

Under these circumstances, the overhead implied by information collection may not worth the effort of a pondered task assignment method. Grids, also known as homogeneous grid environments, have been studied and simulated by projects such as DAMIEN [8, 9]. In a desktop grid the situation is quite different.

2.2 BSP model and extensions

Dealing with a large scale parallel program on the distributed memory architecture which characterizes a desktop grid, it is important to minimize the amount of communication and perform the required communication through reliable channels. In order to do so efficiently, the PRAM model may be dropped in favor of the BSP model for the virtual computer. This model ensures that all communication will complete at the super-step synchronization barrier. Task assignment also takes place at this point. For this reason all communication between individual nodes participating in the super-step is funneled through the assignment server. Construction of a simulator with this model is also quite straightforward and provides a means for verifying the scalability of different task assignment methods. With this tool in hand, performance analysis may be done on several task assignment strategies, such as random, round robin, least-used, greedy communications cost, greedy idle computational cycles and compared with the force field approach.

2.3 Desktop Grid Simulator

Communications cost is defined by several instantaneous network characteristics. Usually the most important ones are latency and network bandwidth. Otero [10], cites five separate variables: latency time, resource contention time, transfer time, WAN contention time and flight time. The first two variables, $T_{Latency}$ and $T_{Resource}$ reflect the actual instantaneous latency between two nodes in the grid. The second two variables, T_{Send} and T_{WAN} reflect the actual instantaneous bandwidth between these any two nodes in the grid. These four are combined to form the actual time the message takes to travel between the hosts, T_{flight} . This last variable depends on the virtual distance between the hosts. Or, in other words, the virtual distance between the hosts can be estimated from the message flight time. It is important to notice that T_{Flight} is determined not when the initial byte arrives at the target, but when the last byte has been received.

When dealing with desktop grids with very large populations, the Law of Large Numbers reveals that the T_{Flight} distribution —and the virtual distance distribution— will converge to a standard normal distribution. This enables us to construct a simulator for a desktop grid which —unlike other simulators

such as DIMEMAS [9], GangSim [5] or GridSim [11]— will have an unstable virtual distance between nodes, as well as varying computational resources. The simulator focuses on these two aspects under the BSP model.

The construction of the simulator is simple, yet obtaining of results is elaborate. The simulator must generate normal distributions for a large number of nodes and tasks. Numerous runs must be performed to obtain data from which statistical results may be analyzed to evaluate the assignment methods under study.

The generated graphs shown in section 4 took several weeks of wall clock time to obtain (using a 200 processor dedicated cluster).

3 Information Retrieval Considerations

In the implementation of task assignment strategies which take into consideration the network and CPU load distribution throughout the grid, one of the most difficult hurdles to overcome is the information retrieval. For this reason a task assignment routine which does not consider these system variables and does a random or round robin assignment *could* be expected to be competitive or even better than a weighted algorithm. In the case of a grid where the remote resources are well known beforehand, such an assignment policy is feasible and effective. But when dealing with the conditions of a desktop grid, with unknown and changing resources at the remote nodes, this policy is prone to error. In the desktop grid a non pondered task assignment runs the risk of tasks being rejected from the remote node because of insufficient resources. Figure 1(a) shows the amount of rejections produced when the number of parallel tasks within the super-step increases. Each rejection implies a loss in time since the information has to travel to and from the remote node. This time must be added to the super-step duration. As can be observed from figure 1(b), the difference in amount of rejections is only marginal when the size of the grid grows to 50000 nodes.

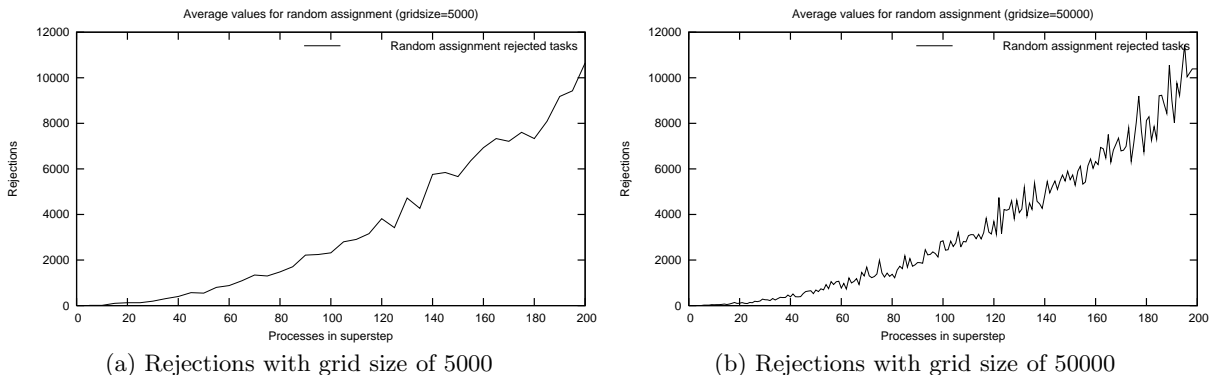


Fig. 1. Task rejections encountered by the random task assignment method

The force field method has been shown to outperform both the greedy communications and greedy idle cycle assignment methods in [4]. But will it outperform blind methods which do not have to worry about collecting information which characterize the grid conditions?

Consider the costs for gathering information in a desktop grid for the force field pondered assignment. Let t_i be the time cost for a information packet to be sent from node i towards the node where the assignment is taking place. Dealing with a desktop grid, the cost for each node i will be different.

Since the force field method determines assignment according to an inverse square law, at the moment of information collection there exists a cut-off time t_c , which may be determined in a desktop fashion. In other words, the information packets arrive at the assignment node, and as they do they are processed. All those information packets which take longer than t_c are ignored.

According to the Law of Large Numbers, it may be assumed that the available remote cycles of the remote processors is normally distributed, as well as the communication costs. This fact allows us to reconstruct the distribution from the partial information which has been received at the assignment node.

Thus, from the information packets that have arrived at the time $t < t_c$ the normal distribution which characterizes the available computation cycles at the remote nodes can be rebuilt and thus the missing values inferred. Whilst these values may be estimated, the actual nodes where they belong cannot be determined, yet this detail is unimportant: the location in grid space of the missing values is irrelevant because they are all more than a certain *distance away* in the force field model. And since force is inversely proportional to the square of the distance, the effect that these have on task assignment quickly disappears. In this manner, with an adequate number of points, the value at the mean of the normal distribution for available computation cycles can be estimated for the desktop grid and the cutoff time can be dynamically obtained by a relatively simple algorithm.

From t_c and the size of the information packets the maximum communication cost can be reconstructed, d_{max} . Thus, for all the nodes for which the information has not arrived, the absolute value of the force field is less than

$$F \leq F_c = \frac{q_1 q_{max}}{d_{max}^2}.$$

where q_1 is the charge associated to the task with greatest memory requirements of the super-step.

Section 4 shows the results obtained by the technique of cutting packet reception after a certain amount of packets have arrived. Due to the previous considerations, according to the size of the grid and the number of tasks to be resolved in the super-step, there will be cut off values where the pondered task assignment methods —such as the force field method— will outperform any random assignment. The experimental results confirm these expectations.

4 Empirical Results

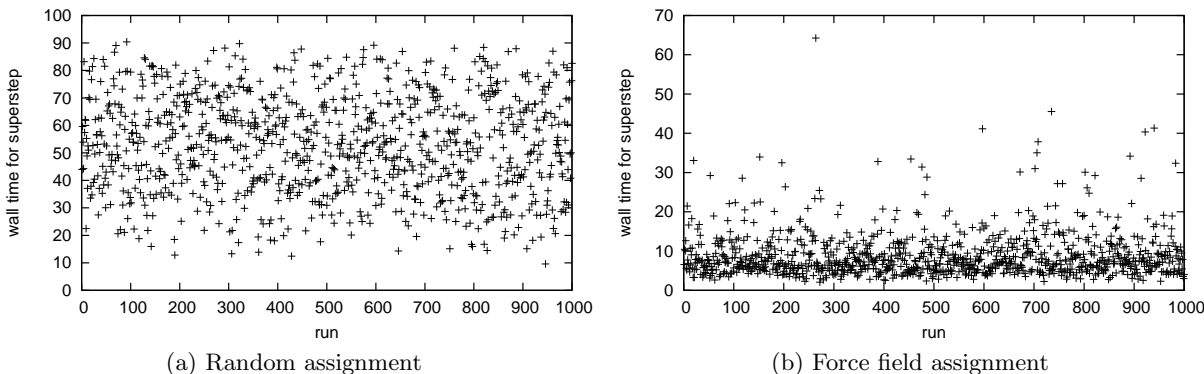


Fig. 2. Super-step execution times in individual runs

Figures 2(a) and 2(b) respectively show the simulation results plotted as point data for the random non-repetitive method and the force field strategy. The force field method in this test considered 20 parallel

threads and a message cutoff after the first 50 messages have been received. The grid population is of 50,000 desktop computers. While from these figures it is clear that the force field is clearly superior, further analysis follows.

Figures 3(a) and 3(b) show the results for two different cutoff values for message reception. In these graphs, the force field method outperforms random assignment. Figure 3(a) shows the results of the the super-step completion times when messages are cutoff after receiving 3 times the quantity of tasks which are to executed within the super-step. A small variation can be observed as the number of tasks increase and the grid size remains constant at 50000. Figure 3(b) show the behavior when the cutoff is raise to 12 times the amount of parallel tasks within the super-step. Not much difference is observed. This indicates that the q_{max} has been surpassed well before receiving three times the number of tasks to execute in parallel.

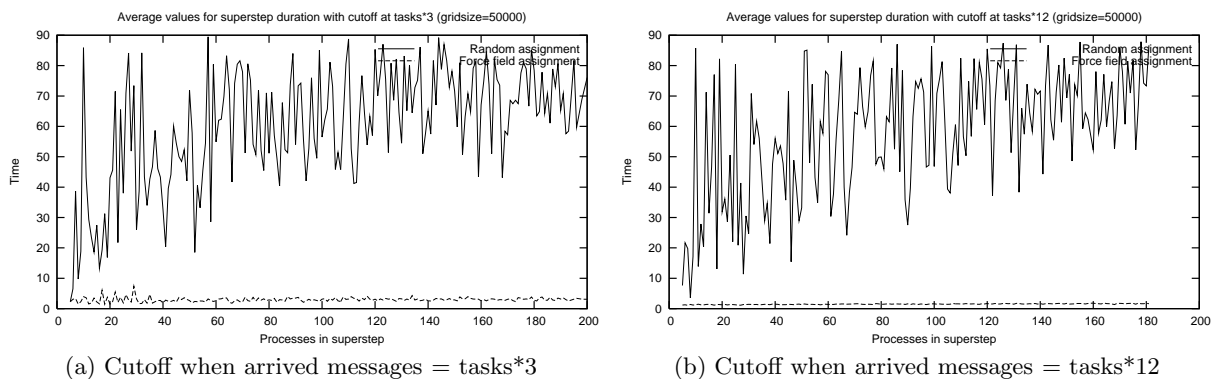


Fig. 3. Super-step execution time for grid size of 50000

5 Conclusions and Future Work

Results show that on large scale grids the force field method outperforms a random task assignment *even* when time loss due to assignment rejections of the blind method is added into the equation. This indicates that the force field method may find application within the arena of grids such as those created by the conjunction of dedicated Linux clusters: more work in this direction is necessary before any refinements to the force field method can be defined for this purpose.

On the other hand, from a practical standpoint, desktop desktop grids are integrated within corporate organizations. Many corporations have a research area which develops new and improved products or services to ensure survival in an ever more competitive market. For the research teams within these organizations, being part of a commercial venture where the economics of cost versus profit determines survival, severe restrictions are imposed on the acquisition of state of the art, high performance computing infrastructure for research. Furthermore, the danger of leaking proprietary information severely hampers the ability to outsource the computational resource for some agreed-upon period of time with some external provider.

In such scenario, nonetheless, the corporation usually has an enormous source of untapped computational potential at its disposal: the amount of desktop computers which are either idle or working well below their thresholds represents an investment with a return value which is not being maximized. The natural tendency —in a business oriented scenario— would be to cover the computational deficit of the research area with the computational surplus located throughout the corporation.

Therein lies the practicality of applying the Force Field task assignment method, which is shown to be an effective approach in this paper. It is shown in both analytic and experimental fashion that the force field model is an effective way to solve the information collection problem for pondered task assignment in desktop grid configurations.

References

1. Dumitrescu, C., Raicu, I., Foster, I.: Di-gruber: A distributed approach to grid resource brokering. In: SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing, Washington, DC, USA, IEEE Computer Society (2005) 38
2. Foster, I., Jennings, N.R., Kesselman, C.: Brain meets brawn: Why grid and agents need each other. In: AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, Washington, DC, USA, IEEE Computer Society (2004) 8–15
3. Wei Fu, Nong Xiao, X.L.: Griddaenfs: A virtual distributed file system for uniform access across multi-domains. In: Proceedings of the 2004 Third International Conference on Grid and Cooperative Computing. Volume 3251., Springer-Verlag (2004) 105–112
4. García, E.W., Morales-Luna, G.: Design of the force field task assignment method and associated performance evaluation for desktop grids. In: Proceedings of the 2005 Fourth International Conference on Grid and Cooperative Computing, Springer-Verlag (2005) 1009–1020
5. Dumitrescu, C.L.: Gangsim: A simulator for grid scheduling studies. In: ACM International Symposium on Cluster Computing and the Grid (CCGRID'05), ACM (2005)
6. et. al., H.B.: Looking up data in p2p systems. *Comm. ACM* **46** (2003) 43–48
7. Zhuge, H.: The future interconnection environment. *IEEE Computer* **4** (2005) 27–33
8. Girona, S., Labarta, J., Badia, R.M.: Validation of dimemas communication model for mpi collective operations. In: EuroPVM/MPI 2000. Volume 1908., Springer-Verlag (2000)
9. Badia, R.M., Labarta, J., Gimenez, J., Escalé, F.: Dimemas: Predicting mpi applications behaviour in grid environments. In: Workshop on Grid Applications and Programming Tools. (2003)
10. Otero, B., Ceá, J.M., Bad'ia, R.M., Labarta, J.: Performance analysis of domain descomposition applications using unbalanced strategies in grid environments. In: Grid and Cooperative Computing – GCC 2005. Volume 3795., Springer-Verlag (2005) 1031–1042
11. Buyya, R., Murshed, M.: Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and Computation: Practice and Experience* **14** (2002) 1175–1220