International Journal of Advanced Research in

Electronics, Communication & Instrumentation Engineering and Development

Volume: 1 Issue: 2 08-Feb-2014,ISSN_NO: 2347 -7210

Investigation and Characteristics Approaches for Nanoscience Materials using networking

Dr.S.Tamil
Associate professor,
Dept.of E.C.E, P.T.Lee.CNCET,
Kancheepuram,

ABSTRACT— *The grid computing is a key to concern when developing parallel operation and distributed computing applications. In this paper the grid management infrastructure is coupled with a performance for local grid load balancing. Each grid scheduler utilises predictive application performance data and an iterative heuristic algorithm to operator. On-demand data streaming is introduced to avoid data overflow using repertory strategies. Experimental results show that balance among task executions with on-demand data streaming is required to improve overall performance Agents cooperate with each other to balance workload in the global grid environment using service advertisement and discovery mechanisms.*

Index Terms: Grid computing, agents, network modeling, Genetic algorithm, Task schedule.

1.INTRODUCTION

In this paper adopts the agent-based methodology to grid load balancing, achieved by coupling the agent system with a performance-driven task scheduler that has been developed for local grid load balancing [1,2]. Each local scheduler uses an iterative heuristic algorithm based on the predictive performance data for each application [3]. The system architecture for grid data streaming pipelines is shown in Figure 1.

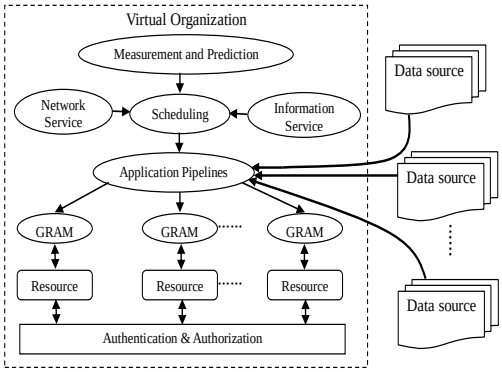


Fig. 1. System Architecture

GM(1,1) is applied to make a prediction of performance of tasks, which provides information for heuristic task scheduling with genetic algorithm [4,5]. Data supply plays a key role in achieving high throughput, and it must be storage aware to avoid data overflow while guaranteeing no task will starve for data. Repertory policy is introduced to control the start and end of data transmissions, and then

storage usage will just be reasonable. To cooperate with the consuming speeds, data transmission rates will be adjusted by tuning parallelism (p) in Grid system [6].

Consider a grid resource P with n processing nodes. A PACE resource model ρ_i can be used to describe the performance information of each processor P_i.

$$P = \{P_i \mid i = 1,2,....., n\} \quad (1)$$

$$\rho = \{\rho_i \mid i = 1,2,....., n\} \quad (2)$$

A set of parallel tasks T is considered to be run on P, where a PACE application model σ_j includes the performance related information of each task T_j. Additionally, T_j is specified with a requirement of the application execution deadline δ_j from the user [7].

$$T = \{T_j \mid j = 1,2,....., m\} \quad (3)$$

$$\sigma = \{\sigma_j \mid j = 1,2,....., m\} \quad (4)$$

$$\delta = \{\delta_j \mid j = 1,2,....., m\} \quad (5)$$

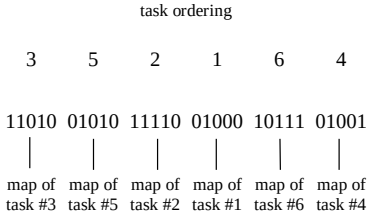
A schedule is defined by a set of nodes $\overline{P_j} \subseteq P$ (corresponding $\overline{\rho_j} \subseteq \rho$) allocated to each task T_j and a start time τ_j at which the allocated nodes all begin to execute the task in unison. The execution time for each task T_j is a function, $t_x(\overline{\rho_j}, \sigma_j)$, provided by the PACE evaluation engine. The completion time η_j of each task T_j is defined as:

$$\eta_j = \tau_j + t_x(\overline{\rho_j}, \sigma_j). \quad (6)$$

The makespan, ω, for a particular schedule, which represents the latest completion time of any task, is subsequently defined as:

$$\omega = \max_{1 \le j \le m} \{\eta_j\}, \quad (7)$$

The goal is to minimise function (7) with respect to the schedule, at the same time $\forall j, \eta_j \le \delta_j$ should also be satisfied as far as possible. In order to obtain near optimal solutions to this combinatorial optimisation problem, the approach taken in this work is to find schedules that meet the above criteria through the use of an iterative heuristic method – in this case a genetic algorithm. The process involves building a set of schedules and identifying solutions that have desirable characteristics. These are then carried into the next generation. The technique requires a coding scheme that can represent all legitimate solutions to the optimisation problem. Any possible solution is uniquely represented by a particular string, S_k, and strings are manipulated in various ways until the algorithm converges on a near optimal solution. In order for this manipulation to proceed in the correct direction, a method of prescribing a quality value (or *fitness*) to each solution string is required. The algorithm for providing this value is called the fitness function f_v.



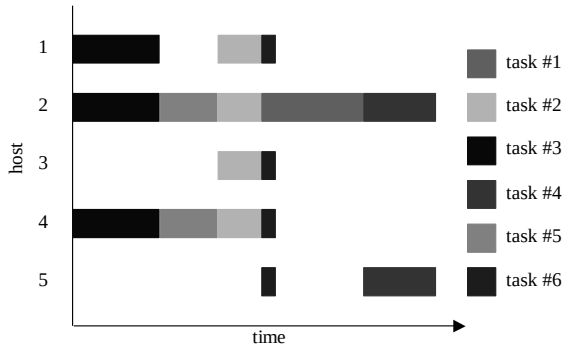


Fig.2. An Example Coding Scheme and Corresponding Gantt Chart

The coding scheme we have developed for this problem consists of two parts: an ordering part, which specifies the order in which the tasks are to be executed and a mapping part, which specifies the allocation of processing nodes to each task. The ordering of the task-allocation sections in the mapping part of the string is commensurate with the task order. An example of a solution string and its associated schedule are shown in Figure 1. The execution times of the various tasks are provided by the performance prediction system and are associated with the task object for evaluation by the fitness function f_v.

A combined cost function is used which considers makespan, idle time and deadline. It is straightforward to calculate the makespan, ω_k, of the schedule represented by any solution string S_k. The nature of the idle time should also be taken into account. Idle time at the front of the schedule is particularly undesirable as this is the processing time which will be wasted first, and is least likely to be recovered by further iterations of the GA or if more tasks are added. Solutions that have large idle times are penalised by weighting pockets of idle time to give φ_k, which penalises early idle time more than later idle time. The contract penalty θ_k is derived from the expected deadline times δ and task completion time η. The cost value for a schedule, represented by a solution string S_k, is derived from these metrics and their impact predetermined by:

$$f_c^k = \frac{W^m \omega_k + W^i \varphi_k + W^c \theta_k}{W^m + W^i + W^c} \qquad (8)$$

The cost value is then normalised to a fitness value using a dynamic scaling technique:

$$f_v^k = \frac{f_c^{\max} - f_c^k}{f_c^{\max} - f_c^{\min}}, \qquad (9)$$

Where f_c^{max} and f_c^{min} represent the best and worst cost value in the scheduling set.

The genetic algorithm utilises a fixed population size and stochastic remainder selection. Specialised crossover and mutation functions are developed for use with the two-part coding scheme. The crossover function first splices the two ordering strings at a random location, and then reorders the pairs to produce legitimate solutions. The mapping parts are crossed over by first reordering them to be consistent with the new task order, and then performing a single-point (binary) crossover. The reordering is necessary to preserve the node mapping associated with a particular task from one generation to the next. The mutation stage is also two-part, with a switching operator randomly applied to the ordering parts, and a random bit-flip applied to the mapping parts. The other possibility is by the movement of the node to the route by overhearing the data packets and processing them for routing information. As shown in figure 3, the intruder node can move to a better position so that the packets can be routed through it as per BSR protocol.

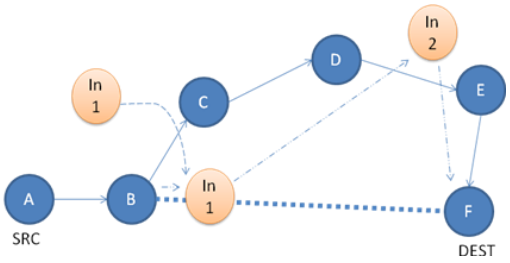


Fig 3. Wormhole Attack by the movement of the node.

Microfabricated chips for DNA analysis and for polymerase chain reaction have already been demonstrated. These are perhaps the initial steps towards a full–fledged technology of biomedical microdevices, which will not only study and analyze nucleic acids but other biological molecules such as proteins, carbohydrates etc. The development of chips for rapid detection of biological pathogens is a critical area with applications in the food handling/processing industry, in biological/chemical warfare, and in early warning for exposure to air and water–borne bacteria, viruses, and other antigens. The µChemLab project is learning to incorporate similar structures into a fully autonomous chemical analysis system that is integrated into on–chip architectures. For example, organically functionalized meso–porous structures have been integrated on a micromachined heating and flow stage to provide 1000–fold chemical pre–concentration for on–chip analysis of chemical warfare agents. Thundat et al. recently demonstrated that the interaction of antigen molecules with their corresponding antibody, attached to the surface of a cantilever, can provide sufficient surface stress to bend the cantilever beam (see Fig. 8.1). Such a device is an example of hybridization and integration at several levels, because it has nanocomponents (antibodies bound to a cantilever surface) and micro components (cantilever beams) which can be delivered in a chip (millimeter scale components) that together integrate biology and biochemistry with engineering. Boxer recently demonstrated the deposition of lipid membranes into lithographically defined corrals. Early results suggest that this approach may allow electrochemical addressing of photo–defined membrane cells. Consequently, this is a first step toward integrating the functional nanotechnology demonstrated by living cells into robust machine architectures. The design and realization of specific structures at near–atomic scales requires controlling materials according to prescribed macro- and meso-scopic specifications, which are actually decided at the quantum level. By comparing the XRD signals of Al-doped and undoped ZnO, it is observed that the addition of aluminum into the films did generate a higher crystallinity. In the present analysis pinpoints spatial variations in the crystallinity for films of similar Al and oxygen contents. The dependency of the electrical resistivity to the cristallinity is in the opposite trend to the observations reported for thin films deposited at a fixed position: the films of better crystallinity exhibit a higher resistivity. The activation of Al dopants can be impaired by the compensation with zinc vacancies or, as we proposed recently , by the formation of the $Al_2O_3(ZnO)_m$ homologous phase.

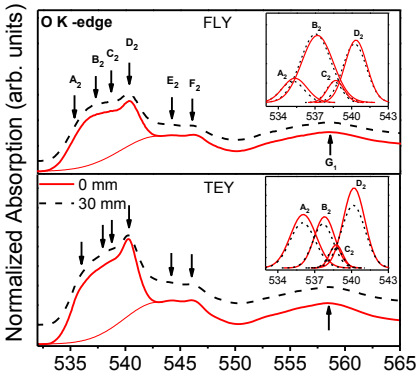


Fig.4 Normal absorption values

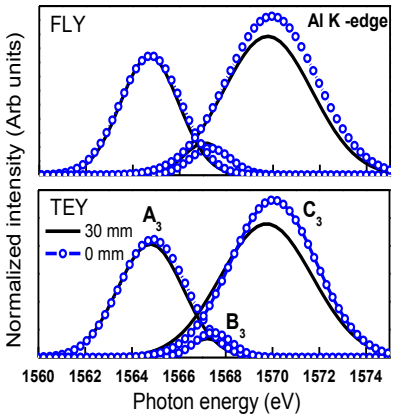


Fig.5. Analysis of normalized intensity

In the Fig.4 FLY and TEY O-K edge XANES of Al-doped ZnO films deposited at 0 and 30 mm fro the magnetron axis and with 5.5 sccm O_2. The right insets were obtained after background subtraction and Gaussian fitting in the low wavelenghts range. But in Fig. 5 the analysis of relevant features of the FLY and TEY Al-K edge XANES of Al-doped ZnO films deposited at 0 and 30 mm from the magnetron axis and with 5.5 sccm O_2 obtained after background subtraction and Gaussian fitting of the spectra system.

In the present work, it was not possible to detect a significant variation of the chemical composition of samples showing very different resistivity values. The main compensation mechanism of Al doping by zinc vacancies, triggered by energetic oxygen ions, is difficult to discard on the basis of the present results. Even, it seems to be obvious if we consider the strong sensitivity of the electrical resisitivity and density of free carriers N to the oxygen flow rate fO_2 during the film growth.

PERFORMANCE ANALYSIS OF MATRIX TASKS

There are a number of performance criteria that can be used to describe grid resource management and scheduling systems. What is considered as high performance depends on the system requirements. In this work three common statistics are investigated quantitatively and used to characterise the effect of grid load balancing. These include: average advance time of application execution completion ε, average resource utilisation rate υ and load balancing level β.

During a period of time t, a set of M parallel tasks T are scheduled onto grid resources P with N processing nodes. Note that the processing nodes may include those at all local grids. The final scheduling scenario can be described using the allocation to each task T_j (with deadline δ_j) a set of nodes $\overline{P_j} \subseteq P$ and a time domain $[\tau_j, \eta_j]$ during which the allocated nodes are simultaneously utilised for task execution.

The average advance time of application execution completion ε can be calculated directly using:

$$\varepsilon = \frac{\sum_{j=1}^{M}(\delta_j - \eta_j)}{M}, \qquad (10)$$

which is negative when most deadlines fail. The resource utilisation rate υ_i of each processing node P_i is calculated as follows:

$$\upsilon_i = \frac{\sum_{\forall j, P_i \in P_j} (\eta_j - \tau_j)}{t}. \qquad (11)$$

The average resource utilisation rate υ of all processing nodes P is:

$$\upsilon = \frac{\sum_{i=1}^{N} \upsilon_i}{N}, \qquad (12)$$

where υ is in the range 0 … 1. The mean square deviation of υ_i is defined as:

$$d = \sqrt{\frac{\sum_{i=1}^{N} (\upsilon - \upsilon_i)^2}{N}}, \qquad (13)$$

and the relative deviation of d over υ that describes the load balancing level of the system is:

$$\beta = 1 - \frac{d}{\upsilon}. \qquad (14)$$

The most effective load balancing is achieved when d equals zero and β equals 1. The three aspects of the system described above are interrelated. For example, if the grid workload is balanced across all the processing nodes, the resource utilisation rate is usually high and the tasks finish quickly. These are illustrated in the case study described in the next section.

The experimental system is configured with twelve agents, illustrated by the hierarchy shown in Figure 6. These agents are named S_1……S_{12} (for the sake of brevity) and represent heterogeneous hardware resources containing sixteen processing nodes per resource. As shown in Figure 6, the resources range in their computational capabilities. The SGI multi-processor is the most powerful, followed by the Sun Ultra 10, 5, 1, and SPARCStation 2 in turn. In the experimental system, each agent maintains a set of service information for the other agents in the system. Each agent pulls service information from its lower and upper agents every ten seconds. All of the agents employ identical strategies with the exception of the agent at the head of the hierarchy (S_1) that does not have an upper agent.

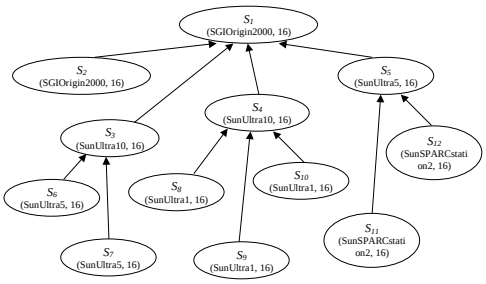


Figure 6. Case Study: Agents and Resources

The applications used in the experiments include typical scientific computing programs. Each application has been modelled and evaluated using PACE. An example of the PACE predications for the system S_1 (which represents the most powerful resource in the experiment) can be found in Table 1.

Table 1. Case Study: Applications and Requirements

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sweep3d [4,200]	50	40	30	25	23	20	17	15	13	11	9	7	6	5	4	4
fft [10,100]	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10
improc [20,192]	48	41	35	30	26	23	21	20	20	21	23	26	30	35	41	48
closure [2, 36]	9	9	8	8	7	7	6	6	5	5	4	4	3	3	2	2
jacobi [6,160]	40	35	30	25	23	20	17	15	13	11	10	9	8	7	6	6
memsort [10,68]	17	16	15	14	13	12	11	10	10	11	12	13	14	15	16	17
cpi [2,128]	32	26	21	17	14	11	9	7	5	4	3	2	4	7	12	20

As shown in the table, different applications have a different performance scenario which has a significant impact on the task scheduling results The required execution time deadline for the application is also selected randomly from a given domain; the bounds of the application requirements can also be found in Table 1. The request phase of each experiment lasts for ten minutes during which 600 task execution requests are sent out to the agents.

LOAD BALANCING

Curves in Figure 9 demonstrate that local and global load balancing are achieved in different ways. While S_1, S_2 and S_{11}, S_{12} are two representative situations, the global situation is not simply an average of local trends as those illustrated in Figures 7 and 8. In the second experiment when the GA scheduling is enabled, the load balancing of local grid resources are significantly improved. In the third experiment, when the agent-based mechanism is enabled, the overall grid load balancing is improved dramatically. It is clear that the GA scheduling contributes more to local grid load balancing and agents contribute more to global grid load balancing. The coupling of both is therefore a good choice to achieve grid load balancing at both local and global levels.

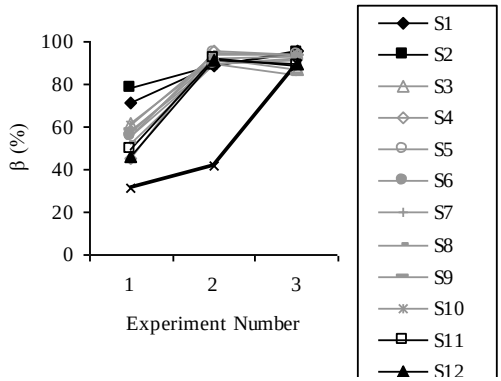


Figure 7. Case Study: Trends III for Experiment Results on Load Balancing Level β

CONCLUSION

In this paper analysis the load balancing in a global grid environment. This is then coupled with an agent-based mechanism that is applied to load balance at a higher level. The technical results demonstrate that the agent-based mechanism coupled with performance-driven task scheduling is suitable for grid load balancing. Load balancing can have large impact on both task executions and resource utilisation. In most situations, good load balancing results in task execution completing earlier with a better utilisation of grid resources. Higher level load-balancing mechanisms coupled with local level scheduling will provide a more effective solution to load balancing on the overall grid than the independent use of either approach.

REFERENCES

[1]. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 2012.

[2]. B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, and S. Tuecke, "Data Management and Transfer in High Performance Computational Grid Environments", *Parallel Computing*, Vol. 28, No. 5, pp. 749-771, 2010.

[3]. I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", *Int. J. Supercomputer Applications*, vol. 11, No. 2, 2010, pp.115-128.

[4] F. Berman, R. Wolski, S. Figueira, J. Schopf, and G. Shao, Application-level scheduling on distributed heterogeneous networks, in "Proc. 2006 Supercomputing", Pittsburgh, PA, USA, 2006.

[5] J. Cao, D. J. Kerbyson, E. Papaefstathiou, and G. R. Nudd, Performance modelling of parallel and distributed computing using PACE, in "Proc. 19th IEEE International Performance, Computing and Communication Conference", pp. 485-492, Phoenix, AZ, USA, 2011.

[6] J. Cao, D. J. Kerbyson, and G. R. Nudd, Dynamic application integration using agent-based operational administration, in "Proc. 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology", pp. 393-396, Manchester, UK, 2010.

[7] J. Cao, D. J. Kerbyson, and G. R. Nudd, High performance service discovery in large-scale multi-agent and mobile-agent systems, *Int. J. Software Engineering and Knowledge Engineering Special Issue on Multi-Agent Systems and Mobile Agents* 5 (October 2009), 621-641.

BIOGRAPHY

Dr.S.Tamil is working as an Associate professor in the Dept.of E.C.E, P.T.Lee.CNCET, Kancheepuram. He has completed his UG in Madras University and PG in Anna University, Chennai. He has 18 years of teaching experience in various reputed institutions. His area of interest are nanoscale structure, time scales analysis and DNA constructs antenna for solar energy