



UvA-DARE (Digital Academic Repository)

Parallel complex systems simulation

Schoneveld, A.

Publication date
1999

[Link to publication](#)

Citation for published version (APA):

Schoneveld, A. (1999). *Parallel complex systems simulation*. [Thesis, fully internal, vg logic/info]. ASCI.

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Chapter 7

Discussion

The goal of this thesis has been to study various aspects of the simulation of complex systems on parallel computers. We have covered only a limited number of issues concerning such simulations. In this final chapter, we will not go into detail discussing the conclusions of the preceding chapters again. However, we will try to resume the main line of thought with which this thesis was originally written and discuss some open issues still remaining.

In the first place, a simulation framework has been defined and implemented, that suits the simulation of arbitrary complex systems on arbitrary parallel computer platforms. Although our definition of arbitrary is more strict than it sounds, i.e., the complex system model somehow has to be defined as a spatially connected task graph, it is general enough for a large class of complex systems. Our framework, called P-CAM, has been designed using the concept of task graphs on the simulation model side and the message passing paradigm on the parallel computer side. In fact it offers a flexible data structure for the parallel simulation of connected, computationally atomic, tasks communicating over mutual links. The P-CAM framework is not intended to be a complete solution for parallel complex systems simulation. It is part of such a solution. It only offers a (dynamic) data structure for a spatial decomposition of a simulation model. We did not consider issues like for example modeling languages and a fully fledged integrated simulation environment. Another very important aspect is the temporal decomposition of the simulation model, which defines the execution scheme. In all but the last chapter, we assumed a synchronous parallel execution scheme, used in most cellular automata (CA) models. Alternative methods are for example event driven schemes. A parallel complex systems simulation environment should also consider such other execution schemes, like parallel discrete event (PDE) methods. It would be interesting to study the interplay and the performance benefit of dynamic load balancing and dynamic time management methods used in PDE (e.g., Time Warp). Such an integrated simulation could be useful in situations where events sparsely occur in time, in a very dynamic heterogeneous domain of spatial activity and inactivity. An example of such a situation is found in individual-based models of population dynamics [104, 160, 161]. In these models, individuals wander around in an environment, where predators may eat prey and prey may eat non-moving food

particles. In many scenarios the prey cluster in groups surrounded by predators. The space between these clusters is empty and as a consequence there is nothing to compute at these locations. However, the clusters themselves are moving and changing in size continuously. Hence, situations may occur where the workload becomes imbalanced if a static partitioning is used. A parallel simulation of such a model using P-CAM in combination with a parallel discrete event method would be an ideal case study.

From the parallel computation side, the optimal allocation of connected tasks, the Task Allocation Problem, is very difficult, and in fact NP-hard. It has been known for quite some time that NP-hard problems show many analogies with a well known class of models in physics, the so called “spin” systems [107]. These “spin” systems are considered to be complex systems, because they are composed of many connected particles (with non-linear interactions). The research even led to the formulation of a physically inspired optimization method called “simulated annealing”. An important property of “spin” systems is that phase transitions can emerge when a certain thermodynamical quantity is varied. In addition it is by now an established fact that that a phase transition is also present in finding optimal solutions to NP-hard optimization problems (see several publications in *Nature* and *Science* [84, 97, 112, 5]). We have found that such a phase transition is also present in the optimal allocations of TAP instances. Several phenomena, like critical scaling and critical slowing down accompany the TAP phase transition. As for the fundamental properties of the general TAP, many issues remain also. Besides random task graphs, task graphs with a specific connectivity should be considered. However, one has to keep in mind that the nice mathematical framework that has been used to study random task graph ensembles is not easily applicable to task graphs with a non-random structure. If the allocation of these non-random task graphs is fully understood, one may wonder how to use these fundamental results in actual task allocation algorithms.

More or less led by its “natural” description, we used the simulated annealing (SA) method to solve the TAP. However, there are many more solutions methods that can be used. We studied a class of such solution methods, called evolutionary algorithms, of which SA is a member. Three different algorithms, instantiated from a uniform parallel optimization framework (implemented in P-CAM), have been compared. It has been established that optimization problems with so called AR(1) energy landscapes are ideal to be explored by SA [157]. Our results indeed indicate that this is true if SA is compared to a Genetic Algorithm (GA) and a Steepest Descent (SD) approach. For GA it is still not known what kind of landscapes it favors. To my opinion this is a very important open problem that needs to be solved. To be able to relate the energy landscape and the “phase” of a specific optimization problem to a particular well suited optimization method, would be a big step forwards in the study of evolutionary algorithms.

Thus far, the more fundamental side of parallel complex systems simulation using P-CAM has gotten all the attention. A more applied research side of P-CAM has also been considered. An issue that has not gotten a lot of attention

in the simulation of complex systems, dynamic load balancing, is studied using P-CAM. A more and more accepted CA based model for computational fluid dynamics, the Lattice Boltzmann Method, in combination with an aggregation process is simulated, while benefiting from the dynamic load balancing capabilities of P-CAM. An other application of P-CAM is its use as a dynamic load balancing support system. This has been the reason to use an existing separate parallel finite element simulation. From the parallel finite element simulation a task graph is deduced, which in turn is used by P-CAM to find optimal task allocations, for a continuously changing workload demand from the several tasks. The optimal use of dynamic load balancing is not yet completely understood. Especially, the important question of “When to balance?” cannot be answered in general. Also, the solutions found by the diffusion based algorithms are not optimal from a task allocation point of view. It is only the workload that is optimally balanced and not the communication load. An “optimal” algorithm, with comparable performance (in order of magnitude) is not known. My feeling is that such an algorithm should be inspired by the method of Molecular Dynamics. A very loose analogy could be made between the interaction forces between particles and the communication loads between tasks. The forces resulting from the communication and the work loads could be used to steer the migration process of the tasks over the processors.

In the last part of this thesis, a different execution scheme is studied (different from synchronous parallel execution) from a complex systems’ point of view. Using an analogy with slowly driven interacting systems, a self organized critical (SOC) state is hypothesized in a specific PDE protocol, called Time Warp. The results are preliminary, and a lot of research has to be done in order to fully understand the Time Warp dynamics and its relation to the SOC state. The complexity of the protocol itself makes this a very difficult task. However, again, together with the TAP phase transition result, this indicates that indeed many problems in the field of parallel computing can be studied as if they were complex systems problems.

