

Making Parallel Processing on Clusters Efficient, Transparent and Easy for Programmers

Andrzej M. Goscinski
School of Computing and Mathematics
Deakin University, Geelong, Victoria 3217, Australia
Email: ang@deakin.edu.au

Many institutions (e.g., universities, banks) are moving towards clusters. These clusters consist of commodity components such as PCs connected by fast networks. However, the single factor limiting the harnessing of the enormous computing power of these clusters for parallel computing is the lack of appropriate software. Present operating systems are not built to support parallel computing – they do not provide services to manage parallelism.

Managing the available parallelism in a cluster means managing parallel processes and computational resources, in order to achieve high performance and use computational resources efficiently, and to make programming and use of the parallel system easy. Parallelism management in parallel programming tools, distributed shared memory and enhanced operating system environments has been in the majority of trials left to application programmers. Programmers must deal not only with programming of communication and coordination of parallel processes to achieve the correct execution of an application, but also with the problems of initialisation and control of the execution on the cluster. Users do not see a cluster as a single powerful computer. Furthermore, parallel systems are seen as being user unfriendly, due to their complexity.

An analysis of the existing clusters shows that parallelism management systems are being developed and offered using two different approaches: middleware, at the application level; and underware, at the kernel level. Both of them have advantages and disadvantages. The problem is how to take advantage of both of them to create the best solution from the point of view of the execution performance, programmers and efficient use of resources.

In the majority of execution environments programmers of parallel application cannot make a choice between the message passing and distributed shared memory communication paradigm. Both paradigms have advantages and disadvantages. The former is fast but difficult to use, the latter is easy to use but demonstrates reduced performance. These communication paradigms and the systems supporting them are treated independently of an operating system, rather than to be parts of a comprehensive operating system as they manage system resources.

Parallel processing can be divided into the initialisation, execution and termination phases. Currently, researchers and manufacturers mainly concentrate their work on the execution phase in order to achieve the best performance. Ease of use of parallel systems and programmer's time are neglected. This approach discourages application programmers from parallel processing, as they have to program many activities, which are of an operating system nature, in particular those of the initialization and termination phases. These activities should be carried out automatically by a cluster operating system to relieve programmers from error prone and time-consuming activities.

In this talk an overview of our work carried out toward the development of cluster operating systems that automatically and dynamically support parallel processing is presented. There are a number of aims of this talk. The first aim is to identify and discuss the basic issues of and solutions to the problem of the management of parallel processing on clusters. The second aim is to propose a new class of cluster operating systems that provide these services. In particular, these operating systems should: guarantee high performance of parallel processing on clusters and the efficient use of resources; support execution on a cluster of both message passing and shared memory based parallel applications; relieve programmers from error prone and time consuming work of allocation of processes to computers, management of interprocess communication and process synchronisation; provide transparency; and make the whole cluster based parallel system easy to use. The third aim is to introduce and discuss the architecture and services of a cluster operating system, called GENESIS, that allow the above specified goals to be achieved. The fourth

aim is to report on the performance of parallel processing of applications executed in the GENESIS environment.