

Operating System Support for Massively Parallel Computer Architectures: An Introduction

MIN-YOU WU¹ AND WOLFGANG SCHRÖDER-PREIKSCHAT²

¹ State University of New York, Buffalo, NY 14260

² GMD FIRST, 12489 Berlin, Germany

The design and development of parallel operating systems is quite a new discipline when compared to that of classical operating systems. At the end of the 1980s, this discipline increased in importance with the breakthrough of parallel computer systems based on distributed-memory architectures. During the 1990s, in order to fulfill the increasing demands of high-performance parallel computing, parallel operating systems will be even more important in the realm of massively parallel computer architectures.

In the past, parallel computing was dominated by two areas: hardware and (numerical) algorithms. The presence of distributed-memory parallel computers raised big challenges not only in these two areas but also in compiler technology. The emphasis in compiler development was—and still is—on automatic parallelization and data partitioning. Closely related to the compiler branch, the demand for new programming models became evident. Nevertheless, the two areas mentioned above still play the dominating role in the first half of the 1990s—operating-system issues are hardly ever addressed.

“A widespread misconception is that the two most important parts of the high performance field are architecture and algorithm.”³

Since the introduction of parallel machines, there has seemingly been no energetic demand to provide specifically designed system software support other than compiler and runtime environments. The approach was (and is) to bypass the operating system—still further, to sacrifice the operating system. There was little enthusiasm for thinking about means by which an operating system for these machines could be designed such that bypassing the operating system never allows the integrity of the kernel and the application tasks to be violated, not to mention discovering operating-system designs with the view that bypassing the operating system could be a possible but never mandatory feature.

In the meantime, this attitude changed. It is very well accepted now that forthcoming (massively) parallel systems must offer a complete solution to achieving high performance, programmability, and supporting availability and management. Modern computer architecture research, thus, means seeking for means by which “seamless transitions” between the various hardware, firmware, and software layers of the system are established.

This special issue contains a collection of the six best articles of the two minitracks “Operating

Received April 1994

Revised May 1994

E-mail may be sent to Min-You Wu at wu@cs.buffalo.edu and to Wolfgang Schröder-Preikschat at wasch@first.gmd.de.

© 1994 by John Wiley & Sons, Inc.

Scientific Programming, Vol. 3, pp. 273–274 (1994)

CCC 1058-9244/94/040273-02

³ H. J. Siegel et al., Report on the Purdue Workshop on Grand Challenges in Computer Architecture for the Support of High Performance Computing. *Journal of Parallel and Distributed Computing*, Vol. 16, pp. 199–211, 1992.

System Support for Massively Parallel Computer Architectures” and “Program Partitioning and Scheduling for Parallel and Distributed Systems” of the 27th Hawaii International Conference on System Sciences (HICSS-27), Maui, January 4–7, 1994. The goals of the minitracks were twofold. First, they tried bringing together researchers of the distributed and the parallel computing area. The idea was to present a forum for discussing

operating-system technology specifically suited for making massively parallel, distributed-memory computers “work fast.” Second, the minitracks aimed at providing a contribution to push forward the design and development of a parallel operating system. The articles presented in this special issue give an overview of the current state-of-the-art of operating-system support for massively parallel computer architectures.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

