DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
UNIVERSITY OF SOUTHERN DENMARK, ODENSE

COMPUTER SCIENCE COLLOQUIUM

Distributed Shared Memory

Brian Vinter
Tromsø University

Tuesday, March 16, 1999, at 2:15 PM
The Seminar Room

Abstract: Conventional wisdom on how to build and program supercomputers is rapidly disintegrating. Machine architecture and programming paradigm are becoming disassociated and portability is becoming a topic even at this level of computation. While nobody can safely say to know what lies ahead in the area of supercomputing certain features are becoming clearer on tomorrows supercomputers. The vector processor is merging with conventional CPU and massively parallel processors are starting to support shared memory as a programming paradigm.

Perhaps the most exiting aspect of this transition is that supercomputers are changing from large custom systems into more commodity parts. This in essence means that you can build your own supercomputer from parts you buy in your local PC store. In fact, for Kr. 500K one can build a 64 CPU supercomputer with a sustained performance close to 20 Gflop/second. Enough to join the list of the worlds top 500 fastest computers.

The problem that arises is then how to program these new supercomputers. Programming with explicit message passing is inherently hard, and any application requires re-orchestration for a new architecture or even a new configuration of an existing architecture. From this point of view, a shared memory programming paradigm preferable. Unfortunately, physical shared memory does not scale well, and anything above 32 CPUs yield rather poor performance. The solution it turns out is to provide a shared memory programming API on a distributed memory architecture.

This talk will introduce and describe the known approaches to distributed shared memory, DSM. We will then go into the details of one such system, the PastSet model, which we are currently working on in Tromsø. The PastSet programming model will be described and it will be shown how a set of applications that use the current PastSet implementation is able to outperform identical applications using explicit messaging systems on the same hardware platform.

Kim Skak Larsen