OpenMP - Cluster Extensions

Thomas Bönisch
boenisch@hlrs.de

SMP - Cluster (Hybrid System)

- Most modern high-performance computing (HPC) systems are clusters of SMP nodes
- DMP (distributed memory parallelization) on the node interconnect
- SMP (symmetric multi-processing) inside of each node
Current Solutions for Programming

- **MPI based:**
  - The MPP model
    - Massively parallel processing
    - Each CPU = one MPI process
  - MPI + OpenMP
    - Each SMP node = one MPI process
    - MPI communication on the node interconnect
    - OpenMP inside of each SMP node
    - DMP with MPI & SMP with OpenMP
  - MPI + automatic parallelization

- **Other models:**
  - HPF, MLP, ...

- **What about an OpenMP-like approach?**

**MPI + OpenMP on SMP-Clusters**

- **Advantages**
  - Could be effective utilizing heavyweight communications between nodes and lightweight threads within a node
  - Less communication packets and uses larger communication packets than pure MPI on SMP Clusters.

- **Disadvantages**
  - Very difficult to start with OpenMP & modify for MPI (non-incremental)
  - Very difficult to program, debug, modify and maintain
  - Generally, cannot do MPI calls from within parallel regions
  - Only people very experienced in both should use this mixed programming model - an intimidating programming model
  - Single Node AND single CPU performance suffers

- Could provide highest performance (at a cost!)
“Distributed OpenMP” Current Approaches

• New directives for distributing data
  – Portland Group (following slides)
  – Compaq

• No extensions to OpenMP itself, but an new more intelligent run time environment
  – Real World Computing Partnership (RWCP) Japan

• KAI announced KAP/Pro Toolset Network Edition
  – presumably also DVSM approach but there is no further information
  – available far 2001

A New Parallel Programming Paradigm for SMP Clusters?

• Ideally, a mixed parallel programming paradigm would exist that
  – enable parallelism using low level, efficient one-way communication between PEs
  – using an efficient multi-threaded mechanism between CPU within a PE

• Such a mixed parallel programming paradigm should
  – allow for full efficiency within an SMP System while also
  – retaining efficiency for distributed memory clusters

• Other desired features:
  – incremental parallelism
  – ease-of-programming
  – maintainability and debugability should be maintained
Distributed OMP

- A merging of SMP and cluster capabilities
  - Merge OpenMP + Craft + HPF technologies
  - Goal is to maintain full source compatibility with OpenMP

- A programmer should be able to
  - start with an OpenMP program and
  - later add mapping directives

Distributed OMP - Characteristics

- Each node is treated as a single process with one-way communication between the nodes

- lightweight threads within a node

- Each node’s program would utilize multiple threads according to OpenMP directives and OpenMP library calls

- Directives for data distribution to the nodes
  - no explicit communication

- Less data movements between nodes plus larger packets of data.
Number of Communications

- 1 CPU x 128 nodes
- 4 CPUs x 32 nodes
- 32 CPUs x 4 nodes

But the message size is increasing in the first case

Distributed OpenMP - Advantages

- One can have a single portable code that can be executed on SMP systems and distributed clusters and hybrid combinations of SMP clusters
- Single SMP node performance will not degrade
- Incremental parallelization capability
- Managers don’t have to risk “betting” on SMP systems being the dominant system
Distributed OpenMP - Disadvantages

- Not yet available (9-12 months)
- More complex than OpenMP or HPF
- User still must give some thought about data partitionment
- Performance tuning after mapping directives are added

“Distributed OpenMP” Current Approaches

- New directives for distributing data
  - Portland Group (following slides)
  - Compaq

- No extensions to OpenMP itself, but an new more intelligent run time environment
  - Real World Computing Partnership (RWCP) Japan

- KAI announced KAP/Pro Toolset Network Edition
  - presumably also DVSM approach
  - but there is no further information
  - available far 2001
Compiler-Directed Software DSM

- CD-SDSM transparently executes OpenMP programs on Clusters of SMP's

- Compilation:
  - The particular compiler compiles the OpenMP C-program to a multi-threaded C-program with runtime library calls.
  - This program is compiled using a native C-compiler
  - linked with the libraries

- The executable is executed on each SMP node in a SPMD fashion (like MPI)

CD-SDSM Internals

- A part of the address space of each process (running on a SMP node) is virtually shared between the nodes.

- Coherence of shared space is maintained by software
  - fine grain coherence control
  - compiler can optimize coherence operations (It knows about the source !!!)
  - these optimizations are crucial to obtain good performance

- Redundant barrier removal
  - Overhead for synchronization is large for SDSM
CD-SDSM Results

- Jacobi Overrelaxation Solver, Matrix 4096x4096, 10 Its.

Summary

- Distributed OpenMP
  - may provide an (excellent?) technology path for SMP Clusters
  - Will be available in 9-12 months
  - Increased complexity
  - Not yet standardized

- CD-SDSM for OpenMP programs
  - available for C
  - shows promising results
  - easier for the programmer

- No experience for large configurations