Multi-Processor Scheduling: Paradigms and Challenges

G. Nelissen, G. Raravi, K. Bletsas, V. Nélis, D. Dasari, A. Burmyakov, P. Souto, E. Tovar

Scheduling Paradigms

Partitioned

- Tasks
- Proc.
- ✓ Reuses the well mature uniprocessor theory
- ✓ Efficient implementation
- ✗ Poor utilization of the resources

Global

- Tasks
- Proc.
- ✓ Good utilization of the resources (up to 100%)
- ✓ Efficient implementation is
- ✓ Task migrations have a cost
- ✗ Increases difficulty and pessimism in timing analysis

Semi-Partitioned

- Partitioned Tasks
- Proc.
- ✓ Trade-off between partitioned and global approaches
- ✓ Implementation is usually a simple variation of partitioned schedulers
- ✓ Good resource utilization

Hierarchical

- Tasks
- Servers
- Proc.
- ✓ Improves task isolation (limit error propagation) at the scheduling level
- ✗ Reduces effective utilization of the resources

A Few Challenges

Task Parallelization

- How can we parallelize code?
- How can we benefit from parallel tasks?
- How can we synchronize the threads if they are mapped on different processors?

Resource Sharing

- Explicit: Shared data or peripherals between tasks
- Implicit: Communication networks, memories, caches, ...
- How can we compute a worst-case access/response/computation time?

Implementation

- Task preemptions/migrations (cache related delays) and the scheduler execution have a cost
- How can the scheduling theory be efficiently implemented?
- How can we back-propagate the actual scheduling overheads to the scheduling theory?

Mixed-Critical & Multi-Mode Systems

- Must go through a certification process
- How can we reduce the design time and cost?
- How can we isolate tasks from different criticality levels?
- How can they safely share resources?

References:


http://www.cister.isep.ipp.pt

clister-info@isep.ipp.pt