

CAPSULE: A Symbiotic Hardware/Software Approach for Facilitating the Parallelization of Programs with Complex Control Flow and Data Structures

Olivier Certner, Zheng Li, Olivier Temam
INRIA Saclay, France

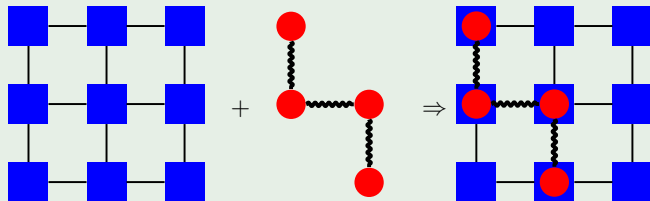
June 5th, 2008

Overview

- Try to provide a simple and intuitive view of the architecture to the programmers
- Be scalable: ultimately aim for distributed-memory model
- Be practical: ride popular software engineering trend = **components**

Multi-Cores + Components

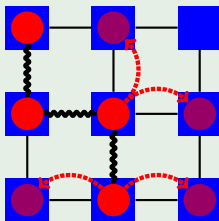
Combining multi-cores and component programs



- Why components ?
 - Decomposition into smaller program parts
 - Isolation of program parts
 - Explicit communication links
- Multi-cores and components blend well together
- Intuitively:
 - Multi-Cores \simeq trade one large core for many smaller cores
 - Components \simeq decompose one large program into many smaller parts

Component Programming Is Not Enough

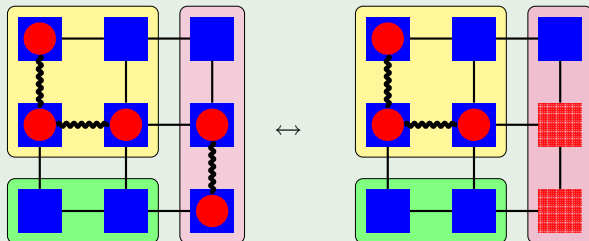
Conditionally dividing components



- BUT component programming \neq finding parallelism
- Augment components with ability to **divide**
- User: **where** divisions occur in program
- System: **when/if** divisions occur \Rightarrow **conditional** divisions (! = *Cilk*)
- System (= architecture and/or run-time) relieves user of granularity, mapping

Marrying Parallelization and Customization Transparently

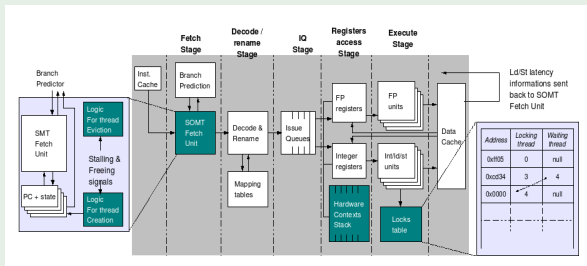
Replacing components with accelerators



- Components transparently mapped to GPP or accelerator
- Run-Time balances resources among applications

A First Implementation On SMTs

SMT with thread division support



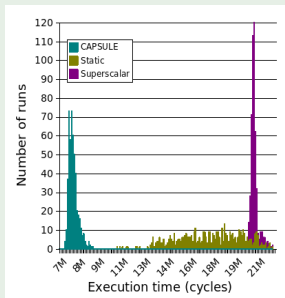
P. Palatin, Y. Lhuillier, O. Temam: Capsule, Hardware-Assisted Parallel Execution of Component-Based Programs, MICRO 2006

- Hardware support for fast thread creation, destruction, swapping.
- 1-cycle probe
- `nthr` = new thread
- Division steering

Extending To Shared-Memory Multi-Processors Without Hardware Support

- No hardware support using **speculative** probe
- Measured 7-cycle probe
- Greater performance predictability \Rightarrow embedded systems

Divisions \Rightarrow performance stability



O. Certner, Z. Li, P. Palatin, O. Temam, F. Arzel, N. Drach: A Practical Approach for Reconciling High and Predictable Performance in Non-Regular Parallel Programs, DATE 2008