universität wien

HPPC @ Euro-Par 2008
August 26, 2008
Las Palmas de Gran Canaria, Spain

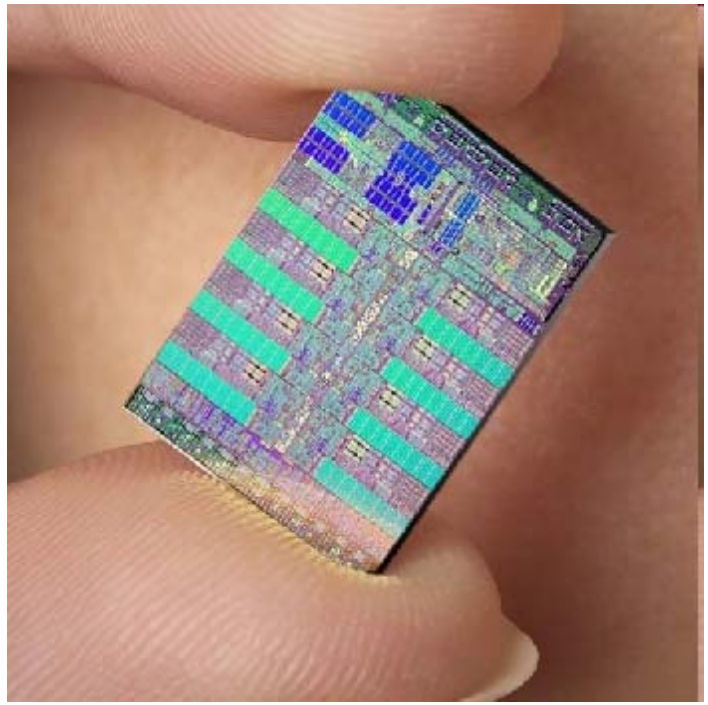
Towards an Intelligent Environment for Programming Multi-core Computing Systems

Sabri Pllana[1], Siegfried Benkner[1], Eduard Mehofer[1], Lasse Natvig[2] and Fatos Xhafa[3]
(1) University of Vienna
(2) Norwegian University of Science and Technology
(3) Polytechnic University of Catalonia

universität wien

Multi-core Processors

- **Alleviate several problems**
  - memory wall
  - power wall
  - ILP wall
- **The price**
  - programmability wall



STI Cell BE

universität wien

Programmability Wall

- **Multi-core programming is complex**
  - shift ILP → TLP
  - low-level native programming
  - lack of suitable tools
- **Portability issues**
  - multi-core architectures differ significantly
- **Lack of skilled programmers**
  - most of programmers are trained for sequential programming

The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information

George A. Miller

Harvard University

Psychological Review, 63, 81-97, 1956

7 ± 2 rule

universität wien

A Chance for Rethinking the Parallel Programming

- **Garage myth**
  - 70's: operating system
  - 90's: search engine
- **HPC market is small**
  - ITC market in 2005 ~ 2000 B€
  - HPC market < 0.5% of ITC market
- **Multi-core pervasiveness**
  - significant part of ITC market
  - affects not only *HPC heroes*
  - time to rethink parallel programming

Sony PlayStation 3

IBM Roadrunner

universität wien

What we propose?

universität wien

An Intelligent Programming Environment

- **Raise the level**
  - of abstraction
  - of automation
- **Higher level of abstraction**
  - model-driven development
  - parallel building blocks
- **Automation & autonomy**
  - intelligent software agents

universität wien

Model-driven Development (MDD)

- **A software development method**
  - *first model* a program, *then build* the program code
- **Inspired by matured engineering disciplines**
  - civil engineering
- **MDD for multi-core programming**
  - capture the program logic as a platform independent model
  - model-based evaluation of functionality and performance
  - reduce the programmer's effort for manual coding
  - generate code for various platforms: the portability issue

universität wien

Parallel Building Blocks (PBBs)

- **Program-independent generic programming units**
  - also known as skeletons or dwarfs
  - support software re-usability
- **Parameters are used to specify the PBB functionality**
  - depend on the context of a certain program
- **PBBs may hide the parallel programming complexity**
  - communication and synchronization
- **PBBs as unit of reasoning in programming environments**
  - useful for code generation, transformation and optimization

universität wien

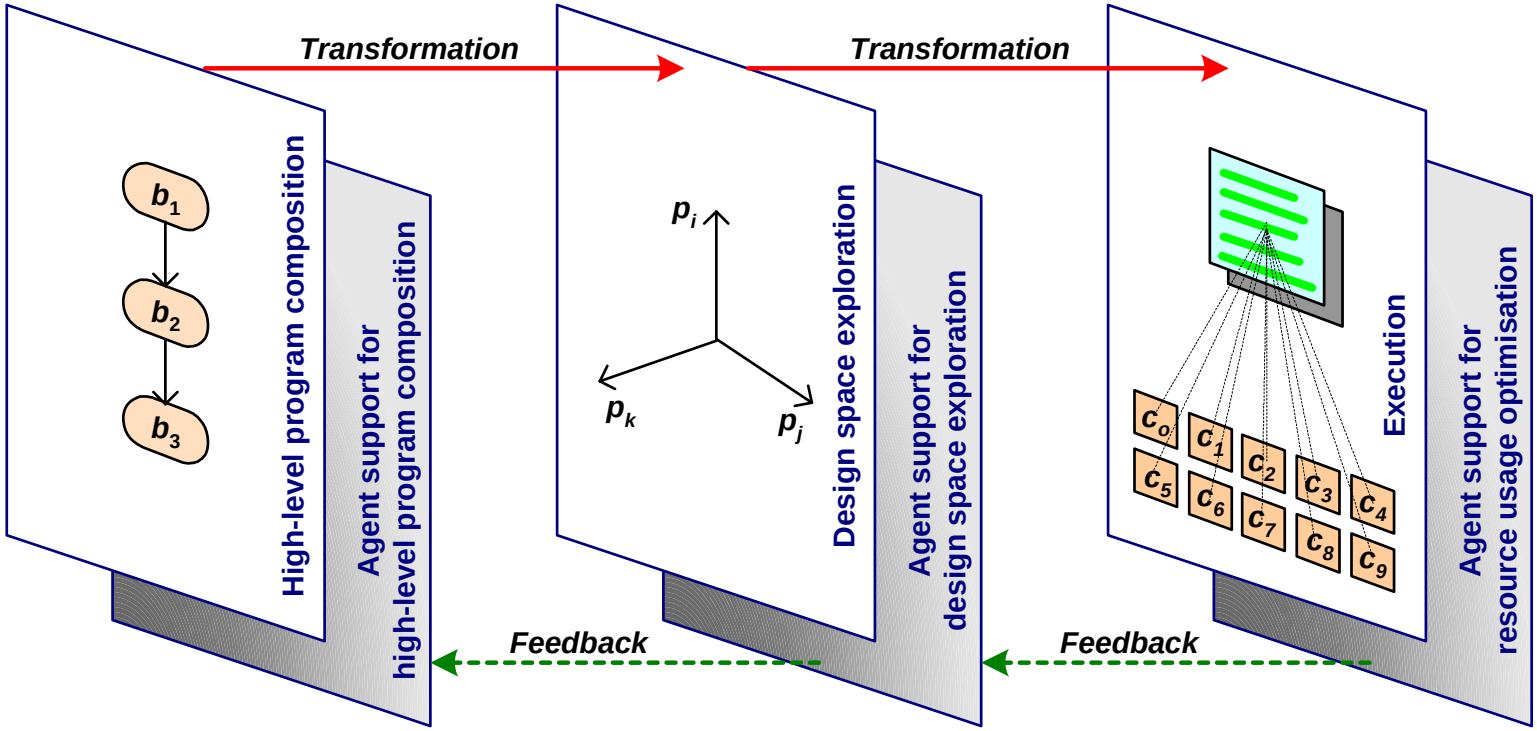
Intelligent Software Agents

- **Intelligent program**
  - *reactive*: responds to changes in the context
  - *proactive*: performs activities based on its initiative
  - *autonomic*: operates independently of the user intervention
  - *social*: communicates and coordinates with other programs

- **Our vision: programming environment supports *proactively* the programmer**
  - program development
  - performance tuning

universität wien

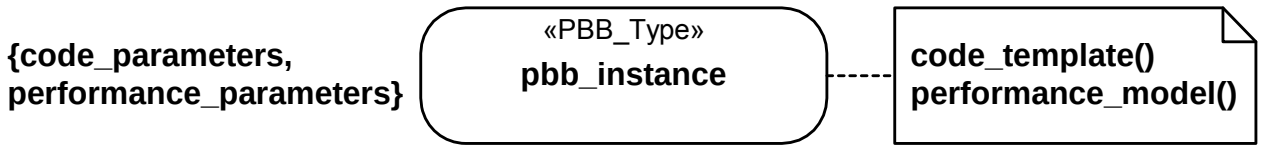
Agent Supported Program Development



universität wien

High-level Program Composition

- **UML-based domain specific language (DSL)**
  - DSL for multi-core systems
- **Parameterized PBBs**
  - code template
  - performance model
- **Proactive composition support**
  - suggest context-appropriate types of PBBs

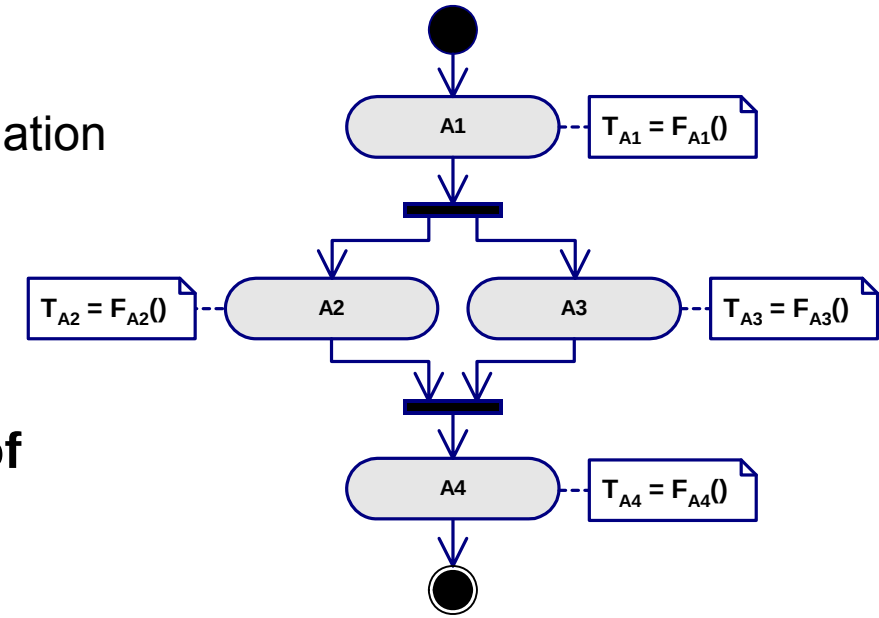


UML representation of a PBB

universität wien

Design Space Exploration

- **A hybrid method**
  - high-level discrete-event simulation
  - mathematical modeling
- **(Semi)-automatic evaluation of various program versions**
  - performance analysis
  - suggest the next steps for the improvement



Hybrid performance model of a hypothetical program

universität wien

Resource Usage Optimization

- **Instruction-level simulation**
  - study the resource utilization (such as on-chip memory)
  - it is a time-consuming activity
  - may be integrated with performance counters

- **Features**
  - runs in autonomic manner in the background
  - propagates the information to the higher-levels of abstraction
  - may be used for calibration of high-level models

universität wien

Experiment

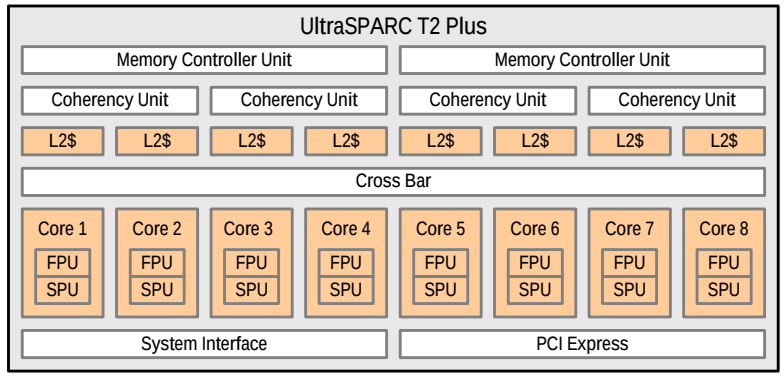
universität wien

Sun UltraSPARC T2 Plus Processor

- **Experimentation platform**
  - Sun Fire T5140
  - comprises two T2+ processors
  - presented by Sun in April 2008

Sun Fire T5140

- **T2 Plus**
  - 8 cores
  - 8 hardware threads/ core
  - 2 integer units/ core
  - 1 FPU/ core
  - 1 Stream Processing Unit (SPU)/ core; cryptographic accelerator
  - 4 MB L2$/ chip



Sun UltraSPARC T2 Plus

universität wien

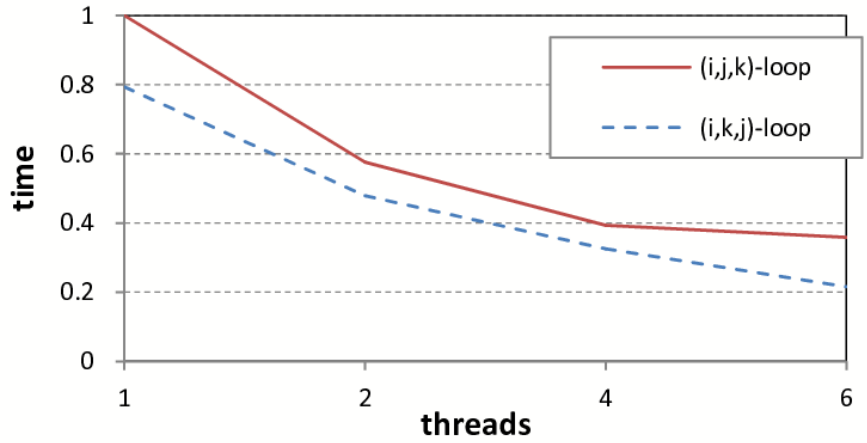
Experiments

- **Floating point matrix-matrix multiplication**
  - C[i,j] = C[i,j] + A[i,k] * C[k,j]
  - loop nest (i,j,k)
- **First parallelization strategy proposed by the programmer**
  - assign C[i,j] to cores in a row-wise manner
  - one thread per core
- **Proactive support of the programming environment**
  - a) detects poor spatial cache locality
  - b) performs loop interchange: (i,j,k) → (i,k,j)
  - c) detects low memory bandwidth and FPU utilization
  - d) suggests the use of hyper-threading

universität wien

Performance Improvement Results

- **Loop-interchange**
  - 1 thread/ core = 26%
  - 2 threads/ core = 20%
  - 4 threads/ core = 20%
  - 6 threads/ core = 66%



Normalized execution times

- **Hyper-threading**
  - (i,j,k) & (i,k,j), 2 threads/ core = 1.7x
  - (i,j,k) & (i,k,j), 4 threads/ core = 2.5x
  - (i,j,k), 6 threads/ core = 2.8x
  - (i,k,j), 6 threads/ core = 3.7x

universität wien

Summary

- **Parallel programming is complex**
  - multi-core pervasiveness is a challenge and an opportunity to rethink parallel programming
- **We have proposed an intelligent programming environment**
  - raises the level of abstraction using MDD and PBBs
  - proactively assists the programmer using intelligent software agents
- **We illustrated the potential benefits with an example**
- **Future work**
  - Realization of the proposed programming environment

Thank you for your attention