



universität
wien

CISIS 2008

March 4-7, 2008
Barcelona, Spain

Hybrid Performance Modeling and Prediction of Large-Scale Computing Systems

Sabri Pilana¹, Siegfried Benkner¹, Fatos Xhafa² and Leonard Barolli³

(1) University of Vienna

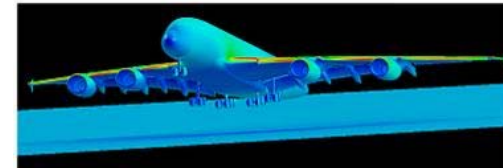
(2) Polytechnic University of Catalonia

(3) Fukuoka Institute of Technology

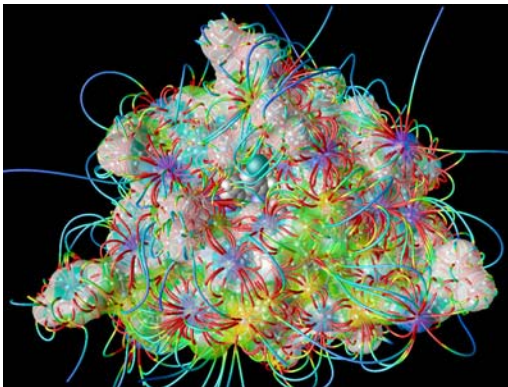


High Performance Computing (HPC) is Essential

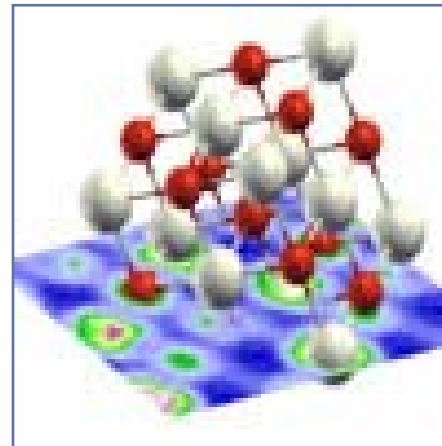
- Engineering
- Scientific discovery
 - Life sciences
 - Material science
 - Earth simulation



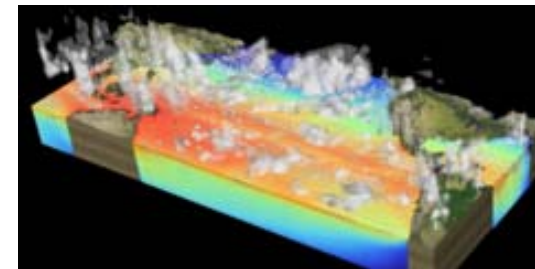
Aerodynamics
© www.dlr.de



Protein folding
© www.research.ibm.com



Material science
© www.wien2k.at



Atmosphere/Ocean
© www.es.jamstec.go.jp



The Program Development for HPC Systems is Complex

- Imperative programming languages
 - Fortran and C
- Programming models
 - Message passing (MPI) and shared memory (OpenMP)
- **Issue:** lack of advanced software engineering methods and tools for HPC
- **Issue:** performance gap
 - Sustained \ll peak
- **Issue:** portability
 - HPC machines about five years
 - HPC programs about 30 years

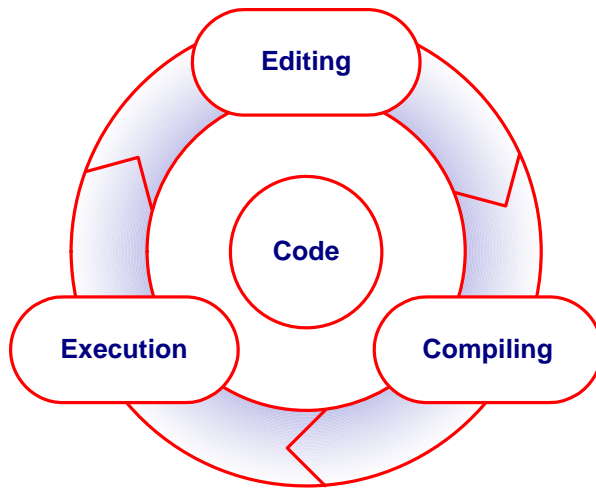


IBM BlueGene/L
© <http://www.llnl.gov>
▶ 212992 processors,
▶ Peak 478 TF/s (Nov. 2007)
▶ 2006 Gordon Bell prize,
Sustained ~57%



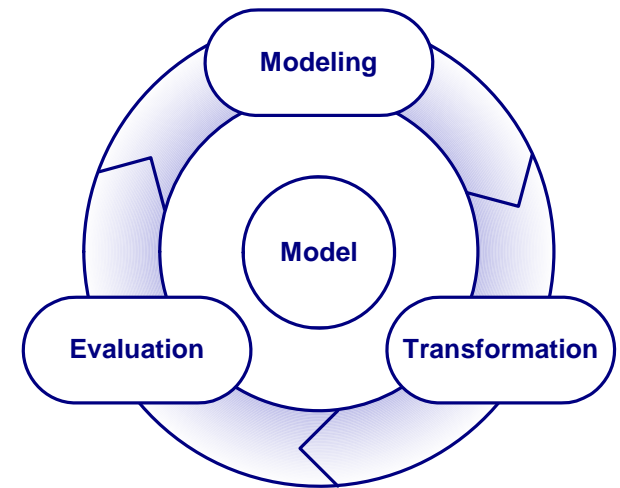
Performance-oriented Program Development

Code-based performance tuning



- ▶ *Time consuming*
- ▶ *Expensive*
- ▶ *Error-prone*
- ▶ *Applicable only to existing systems*

Model-based performance tuning



- ▶ *Efficient*
- ▶ *Inexpensive*
- ▶ *Non-intrusive*
- ▶ *Applicable also to future systems*



Performance Modeling Approaches

■ Mathematical Modeling (MathMod)

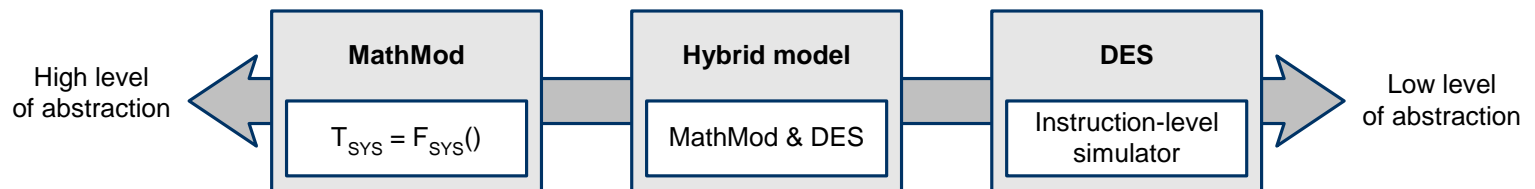
- $T_{\text{ProgExec}} = C_{\text{Op}} T_{\text{Av}}$
- **Fast**, but **not structure-aware**

■ Discrete Event Simulation (DES)

- **Structure-aware**, but **slow**

■ Hybrid Modeling

- Combine MathMod and DES



*Performance model
abstraction levels*

Our Performance Modeling Approach

■ Hybrid performance models

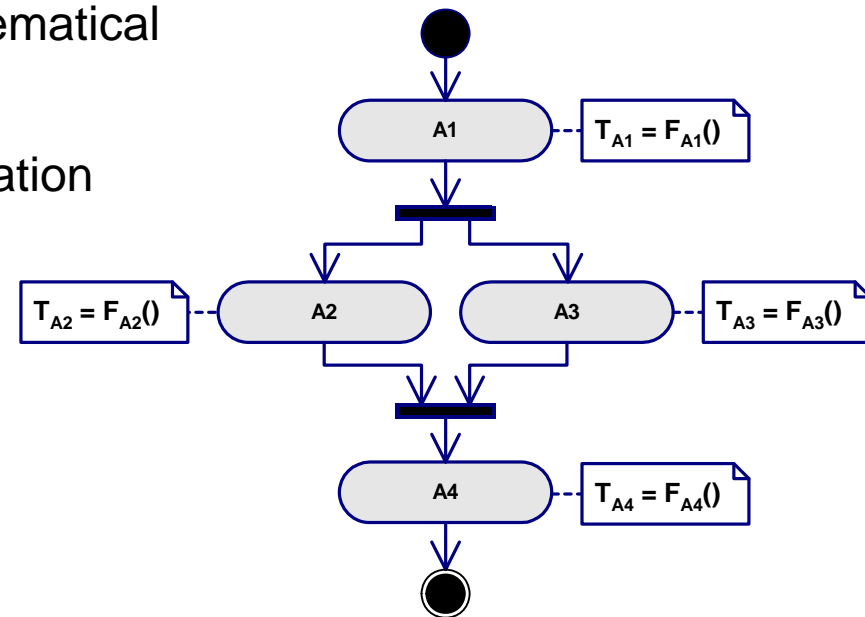
- Evaluation efficiency of mathematical models
- Structure awareness of simulation models

■ Mathematical modeling

- Cost functions represent the performance of activities

■ Discrete Event Simulation

- Control flow

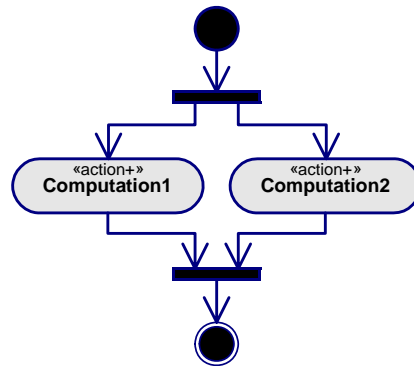


Performance model of a hypothetical program

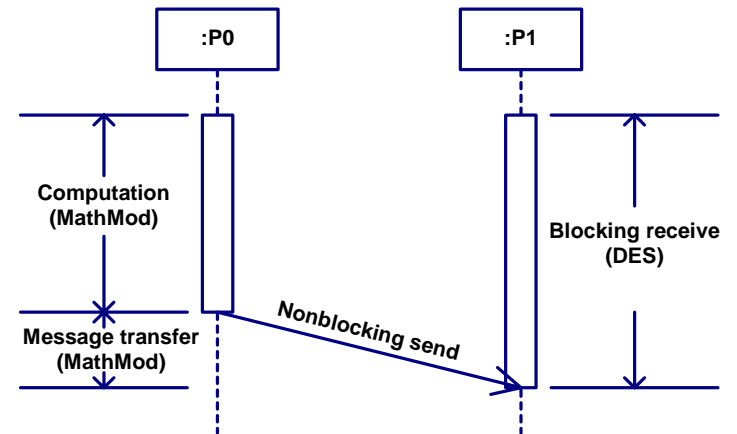
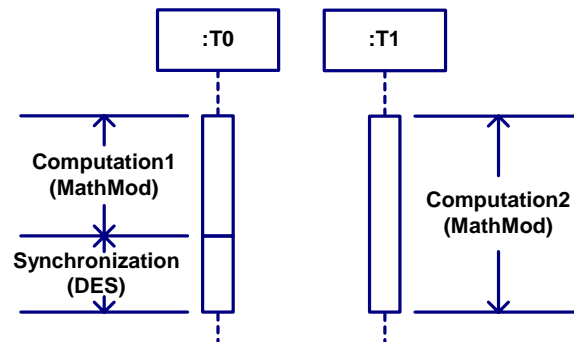
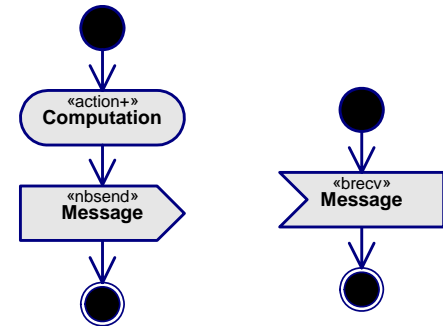


Hybrid Performance Modeling Examples

Parallel region



Inter-process communication





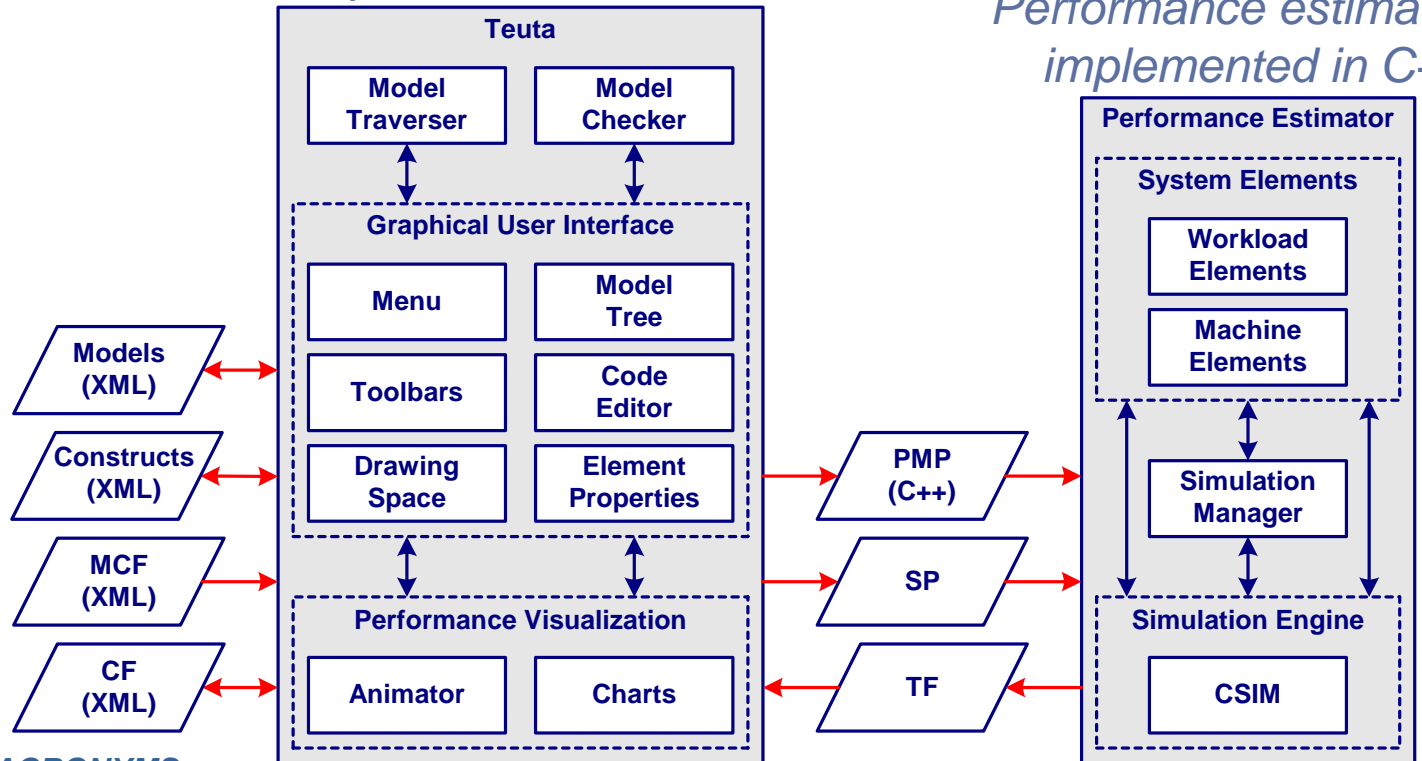
Evaluation



Implementation of Performance Prophet

Teuta is implemented in Java

Performance estimator is implemented in C++



ACRONYMS:

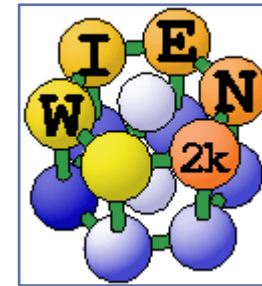
- ▶ MCF: Model Checking File
- ▶ CF: Configuration File
- ▶ PMP: Performance Model of Program

- ▶ SP: System Parameters
- ▶ TF: Trace File
- ▶ XML: Extensible Markup Language

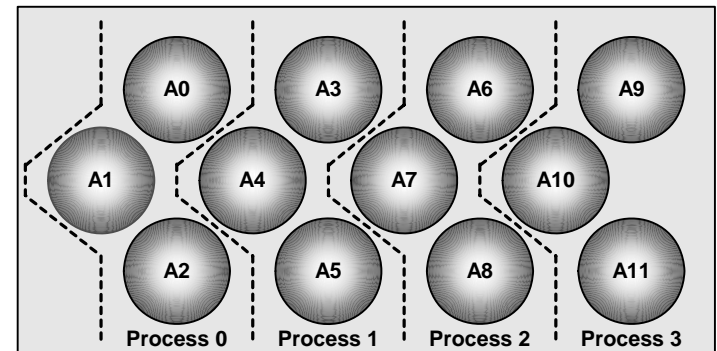


Case Study: Program LAPW0

- **A material science program**
 - Linearized Augmented Plane Wave (LAPW) method
- **Calculates the effective potential within a unit cell of crystal**
- **Written in Fortran 90 and MPI**
 - 15000+ lines of code
- **Developed at Institute of Materials Chemistry, TU Vienna**
 - Prof. Karlheinz Schwarz



*LAPW0 is part of WIEN2k package.
www.wien2k.at*

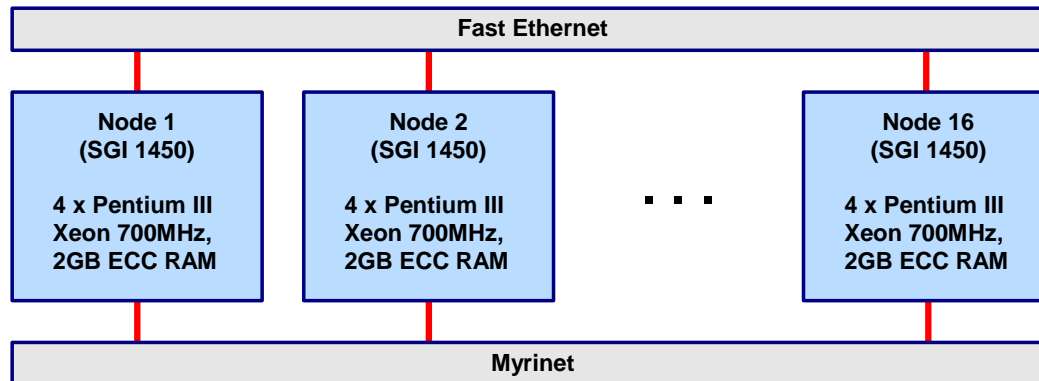


Domain decomposition



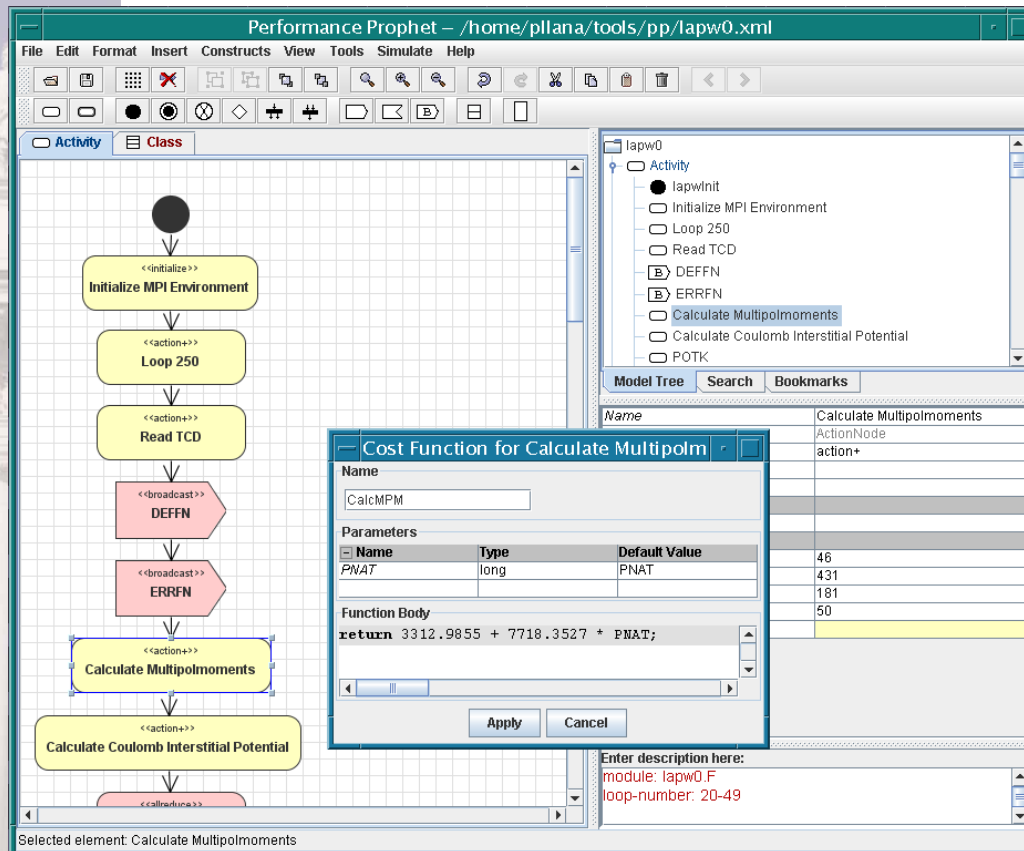
Case Study: Experimentation Platform

- Gescher: a homogeneous cluster of SMP's.
- Located at the Institute of Scientific Computing, University of Vienna





Case Study: Development of LAPW0 Performance Model



We specified the performance model using our UML extension for HPC.

Actions represent code regions that strongly influence the overall performance of LAPW0.

A cost function models the execution time of a code region.



Case Study: Performance Prediction of LAPW0 with Performance Prophet

Visualization of performance prediction results

The system under study is a homogeneous cluster of SMP's.

Specification of system parameters

Simulate Model

Parameters

Number of Nodes: 8

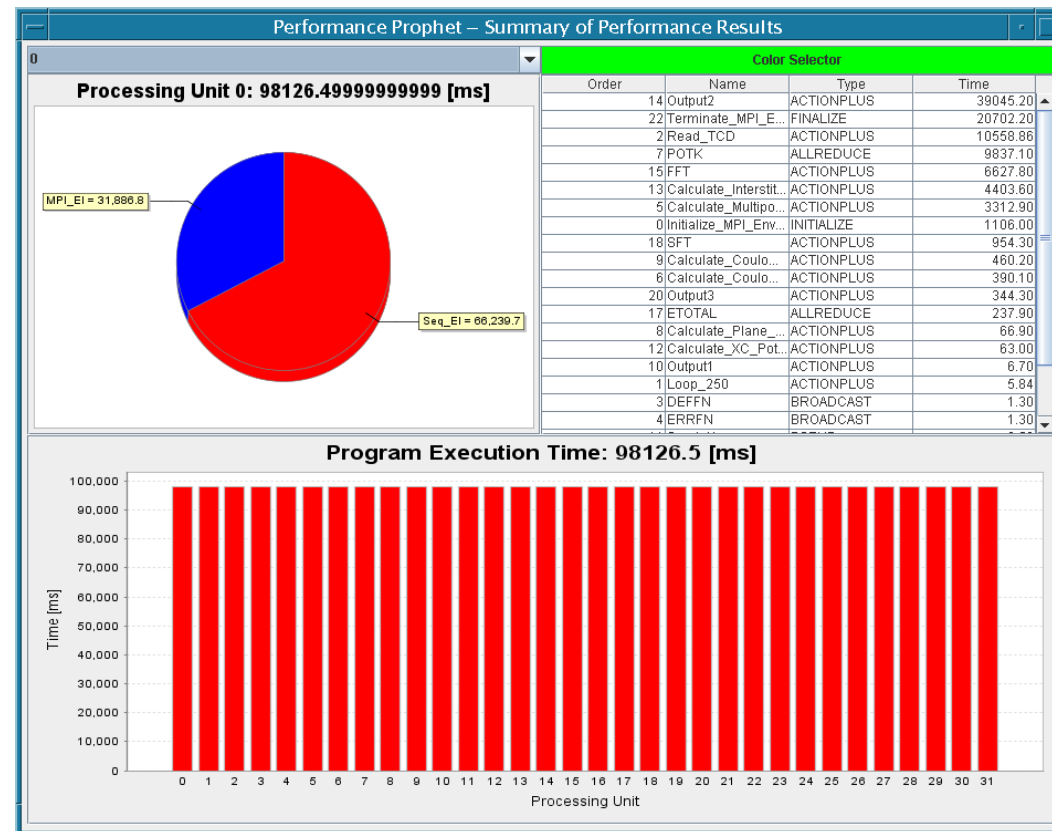
Processors per Node: 4

Number of Processes: 32

Number of Threads: 1

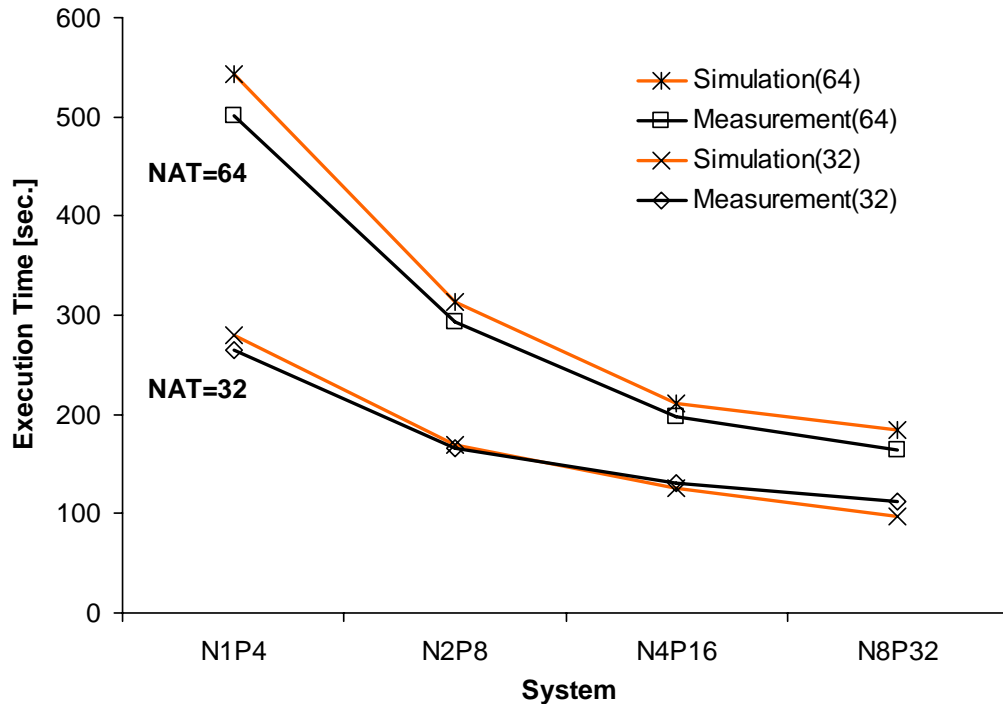
Browse.. /home/plana/tools/pp/experiments/lapw0/tmp

Simulate.. Cancel





Case Study: Validation of the LAPW0 Performance Model



■ T_m : measured execution time of LAPW0 on Gescher

■ T_s : simulated time

■ T_e : model evaluation time

■ Percentage error

$$\square \frac{(|T_s - T_m|}{T_m}) 100$$

■ Average: 7%

■ T_m/T_e : efficiency

$$\square \text{Range: } 1413 - 50100$$



Summary

■ Performance is a key indicator of computing systems

- New computing systems demand new evaluation methods

■ Observations

- Mathematical models: efficient, but lack structural information
- Simulation: structure aware, but applicable to only small programs

■ We presented a hybrid performance modeling approach

- Hybrid: efficient and structure aware
- Empirically evaluated our approach
- Case study: a real-world material science program
- Compared to measurement techniques: > 1000 times faster

■ Future work

- Investigate the applicability of our approach to Grid workflows

Thank you for your attention