Statistical Performance Prediction for Multicore Applications Based on Scalability Characteristics

Oliver Jakob Arndt, Matthias Lüders, and Holger Blume
Outline

- Multicore Performance Prediction
- Scalability Characteristics
- Statistical Prediction Method
- Accuracy Evaluation, Case-Study
Parallel Runtime Behavior

- **Multicores** in all fields
  - Flexible software reduces time-to-market
  - Implementations portable across platforms
Parallel Runtime Behavior

- **Multicores** in all fields
  - Flexible software reduces time-to-market
  - Implementations portable across platforms

- **Parallel programming** requires scalable concurrency
  - Influenced by software demands and hardware capabilities
  - Limited by inappropriate parallelization and bottlenecks
Parallel Runtime Behavior

- **Multicores** in all fields
  - Flexible software reduces time-to-market
  - Implementations portable across platforms

- **Parallel programming** requires scalable concurrency
  - Influenced by software demands and hardware capabilities
  - Limited by inappropriate parallelization and bottlenecks

- **Performance prediction** as supportive tool for developers
Performance Prediction

- **Goal**: Easy, fast, precise prediction

- **System modeling**: Complex in all areas
  - Detailed: modeling effort, simulation
  - Abstract: important effects neglected

![Graph showing performance prediction vs. simulation speed](image)

- **Optimum**
Performance Prediction

- **Goal:** Easy, fast, precise prediction

- **System modeling:** Complex in all areas
  - Detailed: modeling effort, simulation
  - Abstract: important effects neglected

1. **Virtual prototypes:**
   System simulation in software
   - ++ best precision
   - - highest effort
Performance Prediction

- **Goal:** Easy, fast, precise prediction

- **System modeling:** Complex in all areas
  - Detailed: modeling effort, simulation
  - Abstract: important effects neglected

1. **Virtual prototypes:**
   System simulation in software
   - ++ best precision
   - - highest effort

2. **Analytic models:**
   Mechanistic CPU-model, Profiles
   - - moderate accuracy
   - + low modeling effort
Institute of Microelectronic Systems

Performance Prediction

- **Goal:** Easy, fast, precise prediction

- **System modeling:** Complex in all areas
  - Detailed: modeling effort, simulation
  - Abstract: important effects neglected

1. **Virtual prototypes:**
   System simulation in software
   - ++ best precision
   - - highest effort

2. **Analytic models:**
   Mechanistic CPU-model, Profiles
   - - moderate accuracy
   + + low modeling effort

3. **Statistical methods:**
   Machine learning on database
   + + good accuracy
   + + low modeling effort
Prediction with Scalability Characteristics

- **Machine learning approaches**
  - Database design is complex
  - Interfering HW-/SW-features
Prediction with Scalability Characteristics

- **Machine learning approaches**
  - Database design is complex
  - Interfering HW-/SW-features

- **Use of scalability characteristics** (HW-/SW-influences)
  1. Feature extraction from profiles: no modeling effort
  2. Candidate search by distances: no model training
  3. Reconstruction from features: full scalability predicted

- **No user input / architecture-knowledge required**
Scalability Characteristics

- **Scalability**: Capability of spawning work over $n$ cores
  - Denotes bottlenecks and NUMA-/ HT-effects
  - Automatic profiling with MPAL \[1\]

Scalability Characteristics

- **Scalability**: Capability of spawning work over $n$ cores
  - Denotes bottlenecks and NUMA-/ HT-effects
  - Automatic profiling with MPAL \[^{[1]}\]

- **Extracted parameters**:
  - Work imbalance
  - Redundancy
  - Scheduling
  - Lock times

Scalability Characteristics

Scalability: Capability of spawning work over $n$ cores
- Denotes bottlenecks and NUMA-/ HT-effects
- Automatic profiling with MPAL \[1\]

Extracted parameters:
- Work imbalance
- Redundancy
- Scheduling
- Lock times

Characteristics: Represent abstract behavioral perspective (over $n$)

Descriptive Scalability Features

- **Modeled scalability:**
  \[ t(n) = \frac{t(1) \cdot R(n)}{n \cdot (1 - l(n) - w(n) - c(n) - d(n) - s(n) - j(n))} \]

- **Parameters:** Separately modeled
  - Linear base model: two variables
  - Plus linear models for NUMA/HT
  - Curve-fitting returns 6D-vector \( \vec{s}_p \)
Descriptive Scalability Features

- **Modeled scalability:**
  \[ t(n) = \frac{t(1) \cdot R(n)}{n \cdot (1 - l(n) - w(n) - c(n) - d(n) - s(n) - j(n))} \]

- **Parameters:** Separately modeled
  - Linear base model: two variables
  - Plus linear models for NUMA/HT
  - Curve-fitting returns 6D-vector \( \vec{s}_p \)

- **Descriptive vector:** Concatenation
  \[ \vec{s}_c = \begin{bmatrix} \vec{s}_R^T, \vec{s}_l^T, \vec{s}_w^T, \vec{s}_c^T, \vec{s}_d^T, \vec{s}_s^T, \vec{s}_j^T, \vec{pc}^T \end{bmatrix}^T \] (\( \vec{pc} \) – performance counters)

- Quantitative comparison and reconstruction of scaling behavior
Distances and Candidates

- **Database**: Benchmarks $B_i$ profiled on target platforms $T_j$
  - New workload $A$ profiled on reference platform(s) $P$
Distances and Candidates

- **Database**: Benchmarks $B_i$ profiled on target platforms $T_j$
  - New workload $A$ profiled on reference platform(s) $P$

- **Geometric distance**: L2-norm between scaling vectors

- **Candidate selection**: From database
  - Minimum algorithm distance on $P$
  - Minimum platform distance of $B$
Target Scaling Reconstruction

- Interpolating transformation
  - Weighted factors for each element in target scaling vector
  - Variability in database adds to prediction quality
Target Scaling Reconstruction

- Interpolating transformation
  - Weighted factors for each element in target scaling vector
  - Variability in database adds to prediction quality

- Scaling reconstruction
  - Full scaling trend
  - Scaling parameters
  - Performance counters

- Prediction of performance and migration bottlenecks enabled
Accuracy Evaluation

- **17 benchmarks**
  - Real-world algorithms (ADAS) + standard benchmarks
  - Parallelization: domain decomposition, recursive spawns, etc.

- **15 platforms**
  - 6 server-, 6 desktop-, and 3 embedded-processors
  - Varying ages and instruction-set architectures
Accuracy Evaluation

- **17 benchmarks**
  - Real-world algorithms (ADAS) + standard benchmarks
  - Parallelization: domain decomposition, recursive spawns, etc.

- **15 platforms**
  - 6 server-, 6 desktop-, and 3 embedded-processors
  - Varying ages and instruction-set architectures

- **Prediction errors**
  - Server: **25.5 %**, large core-numbers, NUMA+HT
  - Desktop: **9.9 %**, most similarities between cores
  - Embedded: **29.0 %**, too few reference platforms
  - All platforms: **19.9 %**, prediction across processor families
Case-Study

- **Algorithms**: HOG Pedestrian detection, SGM stereo-vision
- **Target platform**: Xilinx Ultrascale+, 4 x ARM Cortex-A53, 1.2 GHz
Case-Study

- **Algorithms**: HOG Pedestrian detection, SGM stereo-vision
- **Target platform**: Xilinx Ultrascale+, 4 x ARM Cortex-A53, 1.2 GHz

- **Virtual prototyping**: GEM5
  - One month modelling
  - 10 h simulation, 16 % error
Case-Study

- **Algorithms:** HOG Pedestrian detection, SGM stereo-vision
- **Target platform:** Xilinx Ultrascale+, 4 x ARM Cortex-A53, 1.2 GHz

- **Virtual prototyping:** GEM5
  - One month modelling
  - 10 h simulation, 16 % error

- **Analytic model:** Exabounds
  - One week modeling, 6 h profiling
  - Prediction in seconds, 25 % error
Case-Study

- **Algorithms:** HOG Pedestrian detection, SGM stereo-vision
- **Target platform:** Xilinx Ultrascale+, 4 x ARM Cortex-A53, 1.2 GHz

- **Virtual prototyping:** GEM5
  - One month modelling
  - 10 h simulation, 16 % error

- **Analytic model:** Exabounds
  - One week modeling, 6 h profiling
  - Prediction in seconds, 25 % error

- **Statistical prediction:** this work
  - 2 h profiling (given database)
  - Prediction in seconds, 19 % error
Institute of Microelectronic Systems

**Conclusion**

- **Statistical multicore performance prediction**
  - Scalability characteristics from profiles: *no modeling required*
  - Simple mathematical model: *no architectural knowledge required*
Conclusion

- **Statistical multicore performance prediction**
  - Scalability characteristics from profiles: *no modeling required*
  - Simple mathematical model: *no architectural knowledge required*

- **Accurate prediction even with small database**
  - Prediction accuracy relies on database
  - Average prediction error < 20 %
Conclusion

- **Statistical multicore performance prediction**
  - Scalability characteristics from profiles: *no modeling required*
  - Simple mathematical model: *no architectural knowledge required*

- **Accurate prediction even with small database**
  - Prediction accuracy relies on database
  - Average prediction error < 20%

- **Easy, fast, and precise multicore-performance prediction**