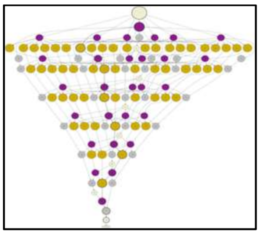
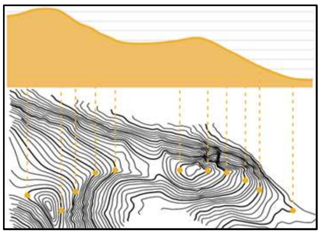
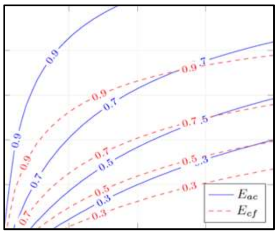
Automatic performance modeling is back in town

Felix Wolf[1], Alexandru Calotoiu[1], Torsten Hoefler[2]
[1]TU Darmstadt, [2]ETH Zurich

TECHNISCHE UNIVERSITÄT DARMSTADT





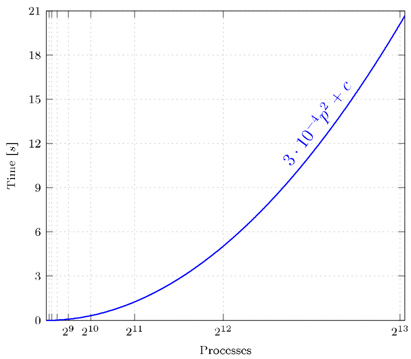


SPPEXA DFG FNSNF

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

Performance model = formula that expresses relevant performance metrics as a function of one or more execution parameters



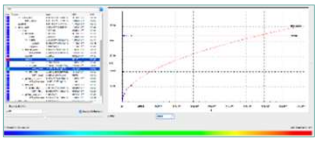
Manual creation challenging

- Identify kernels
  - Incomplete coverage
- Create models
  - Laborious, difficult

Historical background

Catwalk (2013-2017)

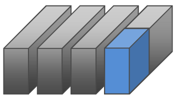
- TU Darmstadt, ETH Zürich, GU Frankfurt
- Goal: automatic empirical performance modeling
- Main result: performance modeling tool **Extra-P**



ExtraPeak (2017-2020)

- TU Darmstadt, ETH Zürich
- Associated with SPPEXA since August 2017
- Goal: allow Extra-P to create models with **multiple parameters**

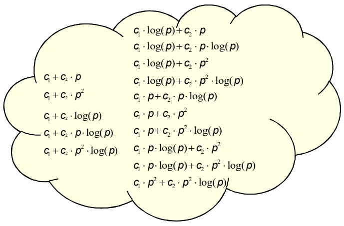
Automatic empirical performance modeling



Small-scale measurements

$$f(p) = \sum_{k=1}^{n} c_k \cdot p^{i_k} \cdot \log_2^{j_k}(p)$$

Performance model normal form (PMNF)



Generation of candidate models and selection of best fit

Kernel [2 of 40]	Model [s] t = f(p)
sweep → MPI_Recv	$4.03\sqrt{p}$
sweep	582.19

While we were away…

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Performance models with multiple parameters
2. Automatic configuration of the search space
3. Segmented models
4. Iso-efficiency modeling

Models with more than one parameter

TECHNISCHE UNIVERSITÄT DARMSTADT

$$f(x_1,..,x_m) = \sum_{k=1}^{n} c_k \prod_{l=1}^{m} x_l^{i_{kl}} \cdot \log_2^{j_{kl}}(x_l)$$

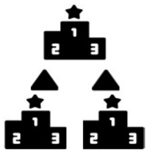
$n = 3$
$m = 3$
$I = \left\{ \frac{0}{4}, \frac{1}{4}, \ldots, \frac{12}{4} \right\}$
$J = \{0, 1, 2\}$

Search space explosion

- Total number of hypotheses to search: 34.786,300,841,019
- Too slow for any practical purpose

Search space reduction through heuristics

TECHNISCHE UNIVERSITÄT DARMSTADT



- **Hierarchical search** – Assumes the best multi-parameter model is created out of the combination of the best single parameter hypothesis for each parameter

- **Modified golden section search** – Speeds up the single parameter search by ordering the hypothesis space and then using a variant of binary search to find the model in logarithmic time rather than linear time

Calotoiu et al.

IEEE Cluster 2016
The Institute of Electrical and Electronics Engineers Cluster 2016
2016.09.13 – 15
Palais de Chine Hotel / Taipei, Taiwan

Automatic configuration of the search space

TECHNISCHE UNIVERSITÄT DARMSTADT

Simplified PMNF

- Constant and “lead order” term

$$f(p) = c_0 + c_1 \cdot p^{\alpha} \cdot \log_2^{\beta} p$$

- c_0 and c_1 are determined by regression
- α and β found automatically via iterative refinement

Results

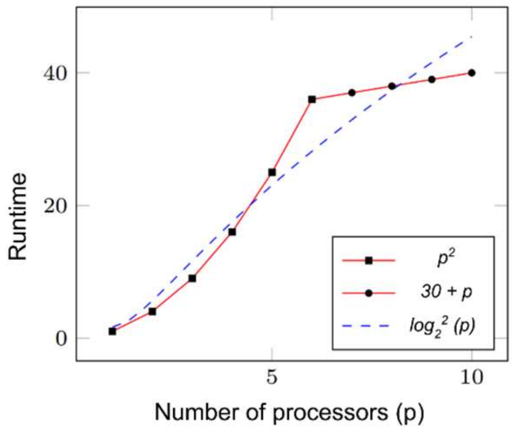
- 4453 models
- 49% remain unchanged
- 39% get better
- 12% get worse
- Improvements in every individual case study

Reisert et al.

Euro-Par 2017

Segmented models

TECHNISCHE UNIVERSITÄT DARMSTADT

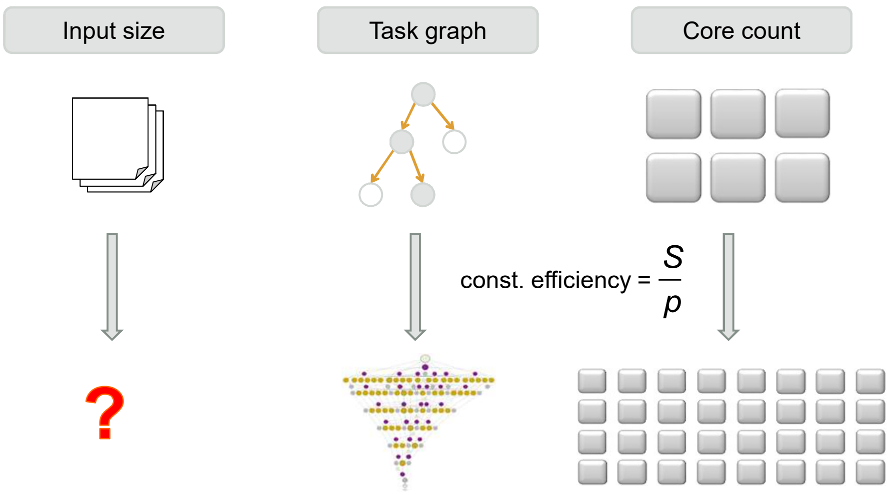


Ilyas et al.

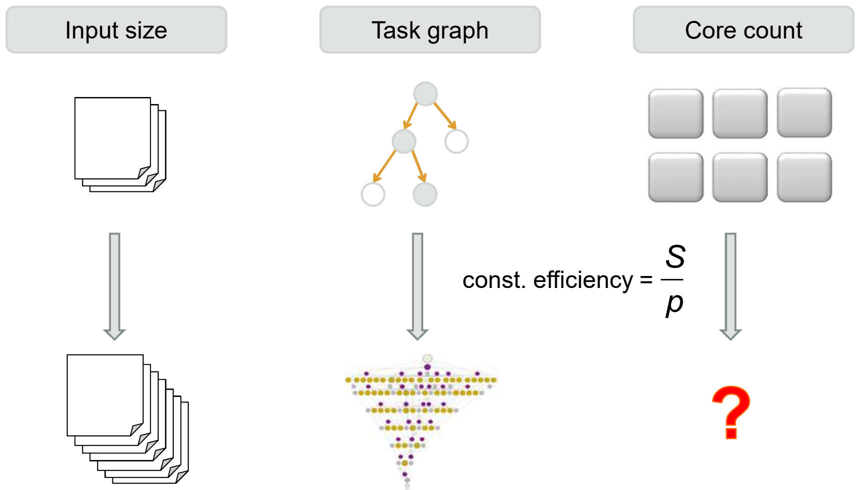
Euro-Par 2017

Efficiency of task-based applications

TECHNISCHE UNIVERSITÄT DARMSTADT



Efficiency of task-based applications



Modeled efficiency functions

TECHNISCHE UNIVERSITÄT DARMSTADT

$E_{ub}(p,n)$ – upper bound based on avg. parallelism

$$\Delta_{str} = E_{ub}(p,n) - E_{cf}(p,n)$$

$E_{cf}(p,n)$ – contention-free replays

$$\Delta_{con} = E_{cf}(p,n) - E_{ac}(p,n)$$

$E_{ac}(p,n)$ – reflects actual performance

Structural discrepancy: Optimization potential at the level of the task graph

Contention discrepancy: Severity of resource contention

Iso-efficiency in co-design

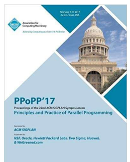
TECHNISCHE UNIVERSITÄT DARMSTADT

App.	Model	Input size for p = 60, E = 0.8
Fibonacci	$E_{ac} = 0.98 - 5.11 \cdot 10^{-3} p^{1.25} + 1.76 \cdot 10^{-3} p^{1.25} \log n$	51
	$E_{cf} = 0.97 - 1.46 \cdot 10^{-2} p^{1.25} + 9.26 \cdot 10^{-3} p^{1.25} \log n$	51
	$E_{ub} = \min\{1, (25.48 + 0.49 n^{2.75} \log n)\, p^{-1}\}$	49
Strassen	$E_{ac} = 1.55 - 1.02 p^{0.25} + 4.59 \cdot 10^{-2} p^{0.25} \log n$	83,600 x 83,600
	$E_{cf} = 1.26 - 0.65 p^{0.33} + 3.89 \cdot 10^{-2} p^{0.33} \log n$	12,680 x 12,680
	$E_{ub} = \min\{1, (0.25 n^{0.75})\, p^{-1}\}$	1,200 x 1,200

For example (Strassen): $E_{ac} = 1.55 - 1.02 p^{0.25} + 4.59 \cdot 10^{-2} p^{0.25} \log n$

Let E = 0.8 and p = 60: $0.8 = 1.55 - 1.02 \cdot 60^{0.25} + 4.59 \cdot 10^{-2} \cdot 60^{0.25} \log n$

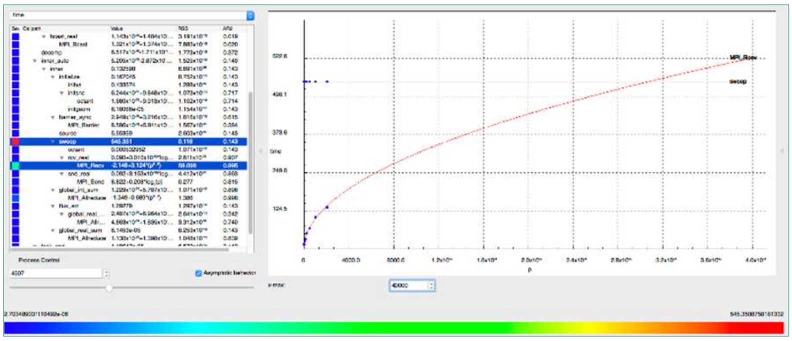
After solving: $n = 83,600$



Shudler et al. PPoPP'17

Extra-P 3.0

TECHNISCHE UNIVERSITÄT DARMSTADT

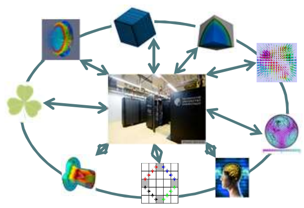


- GUI improvements, better stability, additional features
- Tutorials available through VI-HPS and upon request

http://www.scalasca.org/software/extra-p/download.html

Ongoing work

- Requirements engineering for co-design
  - Applications requirements scale differently for different resources (e.g., network, processor)
  - Goal
    - Model portable requirement metrics
    - Extrapolate to hypothetical system
    - Pay attention to memory locality
- Cost-effective sampling of the performance space
  - Use machine learning to optimize cost or accuracy



Related projects

- TaLPas (BMBF, 2017-2019)
  - Extra-P used to optimize scheduling decisions
  - Partners: TU Munich, U Hamburg, TU Darmstadt, TU Kaiserslautern, U Paderborn, U Stuttgart
- EPE (DFG, 2016-2019)
  - Part of the DFG Program *Performance Engineering for Scientific Software*
  - Extra-P at the center of scalability service
  - Partners: TU Darmstadt, GU Frankfurt, JGU Mainz, TU Kaiserslautern

Publications related to Extra-P

TECHNISCHE UNIVERSITÄT DARMSTADT

[1] Patrick Reisert, Alexandru Calotoiu, Sergei Shudler, Felix Wolf: Following the Blind Seer – Creating Better Performance Models Using Less Information. In Proc. of the 23rd Euro-Par Conference, Santiago de Compostela, Spain

[2] Kashif Ilyas, Alexandru Calotoiu, Felix Wolf: Off-Road Performance Modeling – How to Deal with Segmented Data. In Proc. of the 23rd Euro-Par Conference, Santiago de Compostela, Spain

[3] Sergei Shudler, Alexandru Calotoiu, Torsten Hoefler, Felix Wolf: Isoefficiency in Practice: Configuring and Understanding the Performance of Task-based Applications. In Proc. of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), Austin, TX, USA

[4] Alexandru Calotoiu, David Beckingsale, Christopher W. Earl, Torsten Hoefler, Ian Karlin, Martin Schulz, Felix Wolf: Fast Multi-Parameter Performance Modeling. In Proc. of the 2016 IEEE International Conference on Cluster Computing (CLUSTER), Taipei, Taiwan

[5] Felix Wolf, Christian Bischof, Alexandru Calotoiu, Torsten Hoefler, Christian Iwainsky, Grzegorz Kwasniewski, Bernd Mohr, Sergei Shudler, Alexandre Strube, Andreas Vogel, Gabriel Wittum: Software for Exascale Computing - SPPEXA 2013-2015, chapter Automatic Performance Modeling of HPC Applications.

[6] Andreas Vogel, Alexandru Calotoiu, Arne Nägel, Sebastian Reiter, Alexandre Strube, Gabriel Wittum, Felix Wolf: Software for Exascale Computing - SPPEXA 2013-2015, chapter Automated Performance Modeling of the UG4 Simulation Framework.

[7] Christian Iwainsky, Sergei Shudler, Alexandru Calotoiu, Alexandre Strube, Michael Knobloch, Christian Bischof, Felix Wolf: How Many Threads will be too Many? On the Scalability of OpenMP Implementations. In Proc. of the 21st Euro-Par Conference, Vienna, Austria

[8] Andreas Vogel, Alexandru Calotoiu, Alexandre Strube, Sebastian Reiter, Arne Nägel, Felix Wolf, Gabriel Wittum: 10,000 Performance Models per Minute - Scalability of the UG4 Simulation Framework. In Proc. of the 21st Euro-Par Conference, Vienna, Austria

[9] Sergei Shudler, Alexandru Calotoiu, Torsten Hoefler, Alexandre Strube, Felix Wolf: Exascaling Your Library: Will Your Implementation Meet Your Expectations?. In Proc. of the International Conference on Supercomputing (ICS), Newport Beach, CA, USA

[10] Alexandru Calotoiu, Torsten Hoefler, Felix Wolf: Mass-producing Insightful Performance Models. In *Workshop on Modeling & Simulation of Systems and Applications, University of Washington*, Seattle, Washington, USA

[11] Felix Wolf, Christian Bischof, Torsten Hoefler, Bernd Mohr, Gabriel Wittum, Alexandru Calotoiu, Christian Iwainsky, Alexandre Strube, Andreas Vogel: Catwalk: A Quick Development Path for Performance Models. In Euro-Par 2014: Parallel Processing Workshops

[12] Alexandru Calotoiu, Torsten Hoefler, Marius Poke, Felix Wolf: Using Automated Performance Modeling to Find Scalability Bugs in Complex Codes. In Proc. of the ACM/IEEE Conference on Supercomputing (SC13), Denver, CO, USA