Productivity and Software Development Effort Estimation in High-Performance Computing

Sandra Wienke
IT Center & Chair for High Performance Computing, RWTH Aachen University

SC Doctoral Showcase; Denver, CO, USA; November 2017
Motivation

- Ever increasing demands for computational power
  - Increasing expenses for, e.g., HW acquisition, electrical power, software development

- HPC procurements
  - Quantifiable metric for informed decisions on how to invest available budgets needed

- Comprehensive metric
  - Focus here: German university HPC centers (in production)
  - Advisory council on scientific matters recommends (amongst others) the integration of energy and personnel costs into funding lines for the German National High-Performance Computing (NHR)
  - My approach: productivity model with predictive power
Agenda

- Productivity
  - Single-application Perspective
  - Multi-application Perspective (Job Mix)
  - Sensitivity Analysis

- Total Cost of Ownership

- Software Development Effort
  - Performance Life-Cycle
  - Identification of Impact Factors
  - Quantification of Factor
    “Pre-knowledge”

- Case Study: Aeroacoustics Simulations

- Conclusion

\[
\text{productivity} = \frac{\text{value}}{\text{cost} \ [\€]}
\]

\[
\€ = \text{HW} + \text{energy} + \text{development costs} + ... \\
\text{development effort [days]} \times \text{salary [\€/days]}
\]
Agenda

• Productivity
  – Single-application Perspective
  – Multi-application Perspective (Job Mix)
  – Sensitivity Analysis

• Total Cost of Ownership

• Software Development Effort
  – Performance Life-Cycle
  – Identification of Impact Factors
  – Quantification of Factor
    “Pre-knowledge

• Case Study: Aeroacoustics Simulations

• Conclusion
Productivity

- Economics

\[ \psi(n, \tau) = \frac{\text{scientific outcome}}{\text{total costs [€]}} = \frac{\sum_i r_{app,i}(n, \tau)}{\text{TCO}(n, \tau)} \]

- My adaptations for HPC purposes

\[ \psi = \frac{\text{output}}{\text{input}} = \frac{\text{value}}{\text{cost [€]}} \]

\[ n \quad \text{no. of compute nodes} \quad r_{app,i} \quad \text{no. of application runs} \]

\[ \tau \quad \text{system lifetime} \quad \text{TCO} \quad \text{total cost of ownership} \]

Productivity: Single-application Perspective

- **Value**: Number of simulation code runs of applications $i$
  
  $$ r_{app,i}(n, \tau) \sim \frac{\alpha \cdot \tau}{t_{app,i}(n)} $$

  - In addition: quality weighting factor for high-scaling app.

- **Cost**: Total Cost of Ownership (TCO)
  
  - Per node and per node type

  $$ TCO(n, \tau) = C^{ot}(n) + C^{pa}(n) \cdot \tau $$

  … + HW acquisition + (initial) dev. effort + HW maintenance + energy + SW licenses + …

Productivity: Multi-application Perspective (Job Mix)

- Number of applications
  - Reduction to $m$ „relevant“ applications
  - E.g. tendering process, cluster statistics

- **Value**: Sum of all application runs
  - Comprehensive metric (all kinds of applications)
  - Capacity-based weighting factor $p_i$

\[
\sum_{i=1}^{m} r_{app,i}(n, \tau) \sim \sum_{i=1}^{m} \left( \frac{\alpha \cdot \tau}{t_{app,i}(n_i)} \cdot \frac{n}{n_i} \cdot p_i \right)
\]

- **Cost**: Application-dependent factors
  - Development effort: sum over all apps $i$
  - Power consumption: consider capacity factor $p_i$
Sensitivity of Productivity Model

- Assumptions in model parameters $\rightarrow$ variances in productivity model

- Sensitivity analysis
  - How uncertainty in productivity model can be apportioned to different model parameters
  - Here: simulation- und variance-based global sensitivity analysis (Saltelli et al.)

- Productivity model
  - Only few (well-understood) parameters must be accurately predicted
  $\rightarrow$ Model is robust (within the given conditions)
Agenda

- Productivity
  - Single-application Perspective
  - Multi-application Perspective (Job Mix)
  - Sensitivity Analysis

- Total Cost of Ownership

- Software Development Effort
  - Performance Life-Cycle
  - Identification of Impact Factors
  - Quantification of Factor
    “Pre-knowledge

- Case Study: Aeroacoustics Simulations

- Conclusion
Estimation of Software Development Effort

- Increasing demands for computational power →
  - Increasing HW and SW complexity in HPC
  - Increasing application development effort
- Integration of effort into productivity model becomes more important
  - Quantifiable metric for effort estimation needed

- Used definition of development effort
  - Needed effort for HPC-typical activities such as parallelization, optimization, port of simulation codes, performance analysis, debugging

- Software engineering (SE): software cost models
  - Example: COCOMO II (focus on code size)
  - But: direct applicability of COCOMO II to (investigated) HPC projects not feasible (focus on performance)
Performance Life-Cycle

- Model of relationship of performance and corresponding development effort

\[
\text{effort} = S \cdot f(\text{performance})^R + T
\]

- Method: regression analysis from collected data

- Data collection: human-subject studies
  - Electronic development diary

- Numerous impact factors on software development effort (captured in \(S, R, T\))

Identification of Impact Factors on Effort

- **Key drivers**
  - Aim: focus on most influencing factors first

- **Method**
  - Ranking of factors based on surveys
  - Combine/ eliminate factors with statistical analysis

- **Realization**
  - Start with set of 11 factors & ask for missing factors
  - Here: professional and student developers in HPC
  - Statistical significance testing:
    one-sided Wilcoxon rank test with Holm correction

<table>
<thead>
<tr>
<th>Impact factors on effort (derived by rank sums of 44 surveys)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-knowledge on HW &amp; parallel prog. model</td>
</tr>
<tr>
<td>Code work</td>
</tr>
<tr>
<td>Pre-knowledge on numerical algorithm used</td>
</tr>
<tr>
<td>Parallel prog. model &amp; compiler/ runtime system</td>
</tr>
<tr>
<td>Performance</td>
</tr>
<tr>
<td>Architecture/ hardware</td>
</tr>
<tr>
<td>Tools</td>
</tr>
<tr>
<td>Kind of algorithm</td>
</tr>
<tr>
<td>Code size</td>
</tr>
<tr>
<td>Portability &amp; maintainability over code’s lifetime</td>
</tr>
<tr>
<td>Energy efficiency</td>
</tr>
</tbody>
</table>

Quantification of Impact Factor
“Pre-knowledge”

- **Method: knowledge surveys (KS)**
  - Knowledge questions are not really answered, but participants rate their confidence in answering the questions

- **Realization**
  - 40 questions: parallel computer architecture/programming models, algorithm
  - Distribution of questions by Bloom’s taxonomy
  - Usage of Ø (pre-)KS results in performance life-cycle

A | I am **confident** that I can adequately answer the question for graded test purposes at this time.

B | I can now answer **at least 50%** of the question or know precisely where I can quickly get the information needed and return here in 20 min or less to provide a complete answer for graded test purposes.

C | I am **not confident** to answer the question sufficiently for graded test purposes at this time.
Agenda

• Productivity
  – Single-application Perspective
  – Multi-application Perspective (Job Mix)
  – Sensitivity Analysis

• Total Cost of Ownership

• Software Development Effort
  – Performance Life-Cycle
  – Identification of Impact Factors
  – Quantification of Factor
    “Pre-knowledge

• Case Study: Aeroacoustics Simulations

• Conclusion
Conclusion

- Methodologies to support informed HPC procurements

- Productivity metric for HPC environments
  - Applicability in real-world multi-job setups
  - Robust to errors in assumptions for parameters
  - Used in procurement process of RWTH Cluster CLAIX (2016)

- Software development effort estimation in HPC
  - Focus on performance: performance life-cycle
  - Identification and quantification of key drivers
  - Support of data collection through tools/ material

Outlook

- Continuing data collection with human-subject research
  - Aim: HPC community approach

- Conditional refinement of productivity model
References


