Performance Engineering for Graduate Students: A View from Amsterdam

Ana-Lucia Varbanescu\textsuperscript{1,2} Stephen Nicholas Swatman\textsuperscript{1} Anuj Pathania\textsuperscript{1}
Performance engineering

• Performance Engineering is a **systematic, quantitative** approach to the design and development of software to **meet performance requirements**.

• Software-oriented approach that focuses on **architecture, design, and implementation** choices.

• Provides information needed to **build software** that **meets performance requirements** within **budget**.

Performance engineering requires multi-disciplinary approaches and thorough knowledge in multiple fields!
Systematic Performance Engineering

1. Collect and analyse (user) performance requirements.
2. Understand current performance.
3. Assess feasibility of the requirements.
4. Assess suitable approaches to meet the requirements (including algorithm and/or system (co-)design).
5. Apply tuning and optimization.
6. Assess progress and iterate back to steps 3–5.
7. Analyse and document the process and the final result
1. **Quantify** (using the appropriate tools and methods) performance using the appropriate metric;

2. **Use** and **compare modelling methods**, and **assess** their usefulness

3. **Classify** and **use performance prediction methods**, and assess their usefulness

4. **Design** an empirical performance analysis process for an application, **analyse results**, and **recommend performance improvement solutions**;

5. **Design** and **use** a suitable **model** for **accurate performance prediction** for a given application;

6. **Apply** and **assess** different (existing) **optimization techniques** to parallel and distributed codes;

7. Design, develop, apply, and assess a **complete performance engineering process** for a given application;

8. Use different **performance engineering tools** (e.g., profilers, microbenchmarks and benchmarks, performance counters libraries, etc.).
Course structure

Lectures
Theoretical and empirical concepts.
Combine fundamental methods and tools with modern, state-of-the-art approaches.
Teaches students how to expand their knowledge.

Labs
Link theoretical aspects with processing and tools that facilitate their application in practice
Small-scale assignments, limited coding
Focus on performance analysis, modeling, and prediction

Project
Experience performance engineering for a real case-study application.
Understand the limitations and challenges of the provided methods and tools
Assessment

Exam: 20-25%
Test theoretical knowledge.
Test the understand of methodological aspects of performance engineering.
Augmented with in-class quizzes to stimulate students’ interest in these aspects during lectures.

Assignments: 25-30%
Grade the ability of solving specific aspects of performance engineering.
Reports worth more than coding/tools.
Showcase the practical challenges of performance engineering.

Project: 50%
Assess the ability of students to plan, execute, and document a complete performance engineering project.
Also include a communication aspect, with intermediate and final results.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Stages</th>
<th>Learning Obj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basics of performance</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Code tuning and optimization</td>
<td></td>
<td>✓ ✓</td>
</tr>
<tr>
<td>Roofline model and extensions</td>
<td>✓</td>
<td>✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>Analytical modeling</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>(Micro)benchmarking</td>
<td>✓ ✓ ✓</td>
<td></td>
</tr>
<tr>
<td>Data-driven and stat. modeling</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>Simulation and simulators</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>Perf. counters and patterns</td>
<td>✓ ✓ ✓</td>
<td></td>
</tr>
<tr>
<td>Scale-out to distributed systems</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>Queuing theory</td>
<td>✓ ✓ ✓ ✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>Polyhedral model</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
</tr>
</tbody>
</table>
Assignments

1. The Roofline Model for a simple kernel
   • Applied for sequential code, parallel code, optimized/accelerated code
   • Demonstrate the model can assess the differences

2. Analytical Modeling and Microbenchmarking
   • Design, calibrate, and evaluate analytical models
   • Design/reuse microbenchmarks for calibration

3. Statistical Modeling
   • Choose and use machine learning models
   • Assess their cost and accuracy

4. Performance Counters and Performance Patterns
   • Learn how to collect and understand detailed performance data
   • Use performance patterns to diagnose and solve performance problems
Results

- Total enrolled
- Passing grades
- Evaluation respondents

Year:
- 2017
- 2018
- 2019
- 2020
- 2021
- 2022
- 2023

Students: 0-40
Feedback

- High grades across all categories.
- High praise for the course structure and interconnection between components.
- High workload students spend 20–50% more time than officially allocated.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Firmly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Firmly Agree</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>“The course …”</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taught me a lot</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>17</td>
<td>18</td>
<td>4.5</td>
</tr>
<tr>
<td>Was clearly structured</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>19</td>
<td>13</td>
<td>4.2</td>
</tr>
<tr>
<td>Was intellectually challenging</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>9</td>
<td>25</td>
<td>4.6</td>
</tr>
<tr>
<td>“I acquired, learned, or developed …”</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factual knowledge</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td>4.4</td>
</tr>
<tr>
<td>Fundamental principles</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>16</td>
<td>11</td>
<td>4.2</td>
</tr>
<tr>
<td>Current scientific theories</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>13</td>
<td>9</td>
<td>3.9</td>
</tr>
<tr>
<td>To apply subject matter</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>22</td>
<td>4.8</td>
</tr>
<tr>
<td>Professional skills</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>13</td>
<td>15</td>
<td>4.4</td>
</tr>
<tr>
<td>Technical skills</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>14</td>
<td>9</td>
<td>4.1</td>
</tr>
<tr>
<td>“… helped me understand the subject”</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>16</td>
<td>4.4</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>16</td>
<td>4.5</td>
</tr>
<tr>
<td>Assignment 3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>10</td>
<td>4.1</td>
</tr>
<tr>
<td>Assignment 4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>13</td>
<td>4.4</td>
</tr>
</tbody>
</table>

*Note: In 2017-2018, assignments were evaluated with a single score; these scores have been duplicated across the four separate assignments in this table.*
Lessons learned

1. Performance Engineering is appealing when treated like a puzzle.
   • We appeal to students’ curiosity to understand why applications behave weirdly on different systems.
2. Provide both methods and tools for each part of the course.
   • Students appreciate the theory much better when they can link it to concrete examples.
3. Do not underestimate empirical analysis efforts, especially when experimental design is missing, and/or automation is not properly defined.
   • We spend time and provide many examples on how this should be done correctly and efficiently.
4. Projects stimulate creativity, and students should be allowed exploration time and space.
   • We provide no endline for our projects, and allow students to try different things.
5. Stimulate critical thinking by reporting on both positive and negative results.
   • We grade the process and the actual insights, and not ultimate speed-up or high-accuracy; understanding why and how methods and tools work and fail is fundamental to such a course.
6. This is an intensive course for both teachers and students.
   • We offer a course that students can rely on and apply for their real-life performance engineering projects.