Performance-Influence Models: Prediction, Optimization, Debugging

Norbert Siegmund
Sven Apel

Alexander Grebhahn
Christian Kästner
Even Domain Experts Struggle with Performance!

HPC Experts Meeting at Dagstuhl
Their Problem: Finding the Optimal Configuration for a Given Hardware Platform?

**Binary configuration options**
- **Coarse-grid solver**
  - IP_CG
  - RED_AMG
  - IP_AMG
- **Smoothers**
  - Jac
  - GS
  - RBGS
  - RBGSAC
  - GSAC
  - BS

**Numeric configuration options**
- **Pre-smoothing steps:**
  - 0
- **Post-smoothing steps:**
  - 0

Stencil code
What is the Influence of Configuration Options on Performance?

Binary options

Numeric options

Performance
## Vision: Performance-Influence Models

### Binary configuration options

<table>
<thead>
<tr>
<th>Coarse-grid solver</th>
<th>smoother</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IP_CG</td>
<td>20.5</td>
<td></td>
</tr>
<tr>
<td>RED_AMG</td>
<td>14.3</td>
<td></td>
</tr>
<tr>
<td>IP_AMG</td>
<td>16.0</td>
<td></td>
</tr>
</tbody>
</table>

### Numeric configuration options

<table>
<thead>
<tr>
<th>Pre-smoothing steps</th>
<th>Post-smoothing steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

### Interactions

<table>
<thead>
<tr>
<th>Interactions</th>
<th>Performance Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED_AMG &amp; GS</td>
<td>-4.5</td>
</tr>
<tr>
<td>RED_AMG &amp; GSAC</td>
<td>+50.5</td>
</tr>
<tr>
<td>IP_AMG &amp; RBGS</td>
<td>-17.2</td>
</tr>
<tr>
<td>IP_AMG &amp; BS</td>
<td>+6.0</td>
</tr>
</tbody>
</table>

![Performance Interaction](image)

Determine the influence of configuration options and their interactions and use it for:

- Understanding
- Debugging
- Prediction and optimization

\[
14.3 + 113.3 - 4.5 - 73.9 + 171.7 = 219.9
\]
Learning Procedure

Sampling → Learning → Performance-Influence Model

Performance-Influence Models for Highly Configurable Systems
Sampling Binary and Numeric Options

Structured sampling approaches for the different kinds of options

Exponential number!
Heuristics for Binary-Option Sampling

• Random?
  – Unlikely to select a valid configuration
  – Locally clustered solutions using SAT

• Heuristics
  – Option-Wise (OW): \{ [\text{lock}], [\text{open}], [\text{arrow}], [\text{empty}], \} 
  – Negative Option-Wise (nOW):
    \{ [\text{lock}][\text{open}][\text{arrow}], [\text{lock}][\text{open}][\text{arrow}], [\text{lock}][\text{open}][\text{arrow}], [\text{lock}][\text{open}][\text{arrow}], [\text{lock}][\text{open}][\text{arrow}] \}
  – Pair-Wise (PW) : \{ [\text{lock}][\text{open}], [\text{open}][\text{arrow}], [\text{arrow}][\text{empty}], [\text{lock}][\text{arrow}], [\text{lock}][\text{empty}], \}
Numeric-Option Sampling (Experimental Designs)

- Fractional factorial designs
- Optimal designs

Pre-study:

Plackett-Burman Design as best design

Multi-grid solver as subject systems
**Plackett-Burman Design (PBD)**

- Minimizes the variance of the estimates of the independent variables (numeric options)
- …while using a limited number of measurements
- Design specifies *seeds* depending on the number of experiments to be conducted (i.e., configurations to be measured)

---

**Value range of a numeric option**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>Center</td>
<td>Max</td>
</tr>
</tbody>
</table>

**Configurations**

<table>
<thead>
<tr>
<th></th>
<th>$O_1$</th>
<th>$O_2$</th>
<th>$O_3$</th>
<th>$O_4$</th>
<th>$O_5$</th>
<th>$O_6$</th>
<th>$O_7$</th>
<th>$O_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$c_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$c_3$</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$c_4$</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$c_5$</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$c_6$</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$c_7$</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$c_8$</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$c_9$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Learning Procedure
## Regression Learning

<table>
<thead>
<tr>
<th>Configurations</th>
<th>Individual Options</th>
<th>Interactions</th>
<th>Functions</th>
<th>Perf.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 0 20 0</td>
<td>0 0 0 0</td>
<td>400 N/A</td>
<td>833</td>
</tr>
<tr>
<td></td>
<td>0 1 1 50 16</td>
<td>0 0 0 800</td>
<td>2500 1.2</td>
<td>411</td>
</tr>
<tr>
<td></td>
<td>1 0 1 100 32</td>
<td>0 0 0 0</td>
<td>10000 0.9</td>
<td>290</td>
</tr>
<tr>
<td></td>
<td>1 1 0 50 32</td>
<td>1 0 1600</td>
<td>2500 1.5</td>
<td>799</td>
</tr>
<tr>
<td></td>
<td>1 1 1 20 32</td>
<td>1 1 640</td>
<td>400 1.5</td>
<td>753</td>
</tr>
<tr>
<td></td>
<td>0 0 0 100 0</td>
<td>0 0 0 0</td>
<td>10000 N/A</td>
<td>514</td>
</tr>
<tr>
<td></td>
<td>1 1 1 100 16</td>
<td>1 1 1600</td>
<td>10000 1.2</td>
<td>416</td>
</tr>
</tbody>
</table>

### Functions

- $\log(\text{Perf.})$

### Individual Options

- $\log(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$

### Interactions

- $\log(\beta_6, \beta_7, \beta_8)$

### Functions

- $\log(\beta_9, \beta_{10})$

### Regression Learning

<table>
<thead>
<tr>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
<th>$\beta_4$</th>
<th>$\beta_5$</th>
<th>$\beta_6$</th>
<th>$\beta_7$</th>
<th>$\beta_8$</th>
<th>$\beta_9$</th>
<th>$\beta_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>102.4</td>
<td>84.3</td>
<td>54.1</td>
<td>5.4</td>
<td>1.3</td>
<td>-14.1</td>
<td>-5.4</td>
<td>2.4</td>
<td>2.1</td>
<td>8.8</td>
</tr>
<tr>
<td>132.3</td>
<td>81.3</td>
<td>56.6</td>
<td>3.5</td>
<td>1.9</td>
<td>-14.1</td>
<td>-5.4</td>
<td>2.4</td>
<td>2.1</td>
<td>8.8</td>
</tr>
<tr>
<td>130.3</td>
<td>83.5</td>
<td>54.2</td>
<td>0.01</td>
<td>0</td>
<td>-14.1</td>
<td>-5.4</td>
<td>1.4</td>
<td>2.1</td>
<td>8.8</td>
</tr>
</tbody>
</table>

### Performance Influence Models for Highly Configurable Systems

- Exponential number!
- Unlimited candidates!
Multiple Regression with Feature Subset Selection

- Extend model in a stepwise manner
- Probe different candidates
  - Individual influences, interactions, functions

Siegmund et al., FSE’15
Learning Procedure

Sampling → Learning → Performance-Influence Model

Performance-Influence Model

```
189392.754871324 * root + 5.341.1497460973 * cells +
1097.33668550875 * post + 6852.5884693721 * GradientSolver +
752.542387648181 * BICGSTABSolver + 1.88645994097469 * cells * post +
547.236697276589 * GradientSolver * pre + pre +
113.547148735501 * cells * GradientSolver + 0.706383967811375 * cells +
cells * cells + 385.8402181761216 * GradientSolver * post + post +
cells * cells + 272.6707363219899 * cells * pre + 39.962334861407 * BICGSTABSolver +
2.72707363219899 * cells * pre + 11.970128727527 * cells * GradientSolver * pre + pre +
pre * pre + 11.970128727527 * cells * GradientSolver * post + post +
8.2288112361045 * cells + 1122.522292956782 * post * SeqIOR
```
Experimental Evaluation

Synthetic → Prerequisite
1. Exp

Real world → Accuracy
2. Exp

Real world → Understanding
3. Exp → use cases
4. Exp → domain knowledge
1. Experiment: Finding the Actual Influences

RQ: Do we find the actually existing influences and interactions?

• Design:

Real systems → Performance model containing only binary options → Re-learned model → Performance model with synthetic numeric options → Ground truth

Sampling

Performance-Influence Models for Highly Configurable Systems
1. Experiment: Finding the Actual Influences

RQ: Do we find the actually existing influences and interactions?

- Design:

  - Real systems
  - Performance model containing only binary options
  - Performance model with synthetic numeric options

  We identified the most relevant influences
  Average prediction accuracy: 98.5 %!
2. Experiment: Performance Prediction

RQ: Which combination of sampling approaches achieve the highest prediction accuracy?

<table>
<thead>
<tr>
<th>System</th>
<th>Domain</th>
<th># Binary Opt.</th>
<th># Numeric Opt.</th>
<th># Constraints</th>
<th># Configs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dune MGS</td>
<td>Multi-Grid Solver</td>
<td>8</td>
<td>3</td>
<td>20</td>
<td>2304</td>
</tr>
<tr>
<td>HIPA&lt;sup&gt;cc&lt;/sup&gt;</td>
<td>Image Processing</td>
<td>31</td>
<td>2</td>
<td>416</td>
<td>13485</td>
</tr>
<tr>
<td>HSMGP</td>
<td>Multi-Grid Solver</td>
<td>11</td>
<td>3</td>
<td>45</td>
<td>3456</td>
</tr>
<tr>
<td>JavaGC</td>
<td>Runtime Env.</td>
<td>12</td>
<td>23</td>
<td>4</td>
<td>$10^{31}$</td>
</tr>
<tr>
<td>SaC</td>
<td>Compiler</td>
<td>53</td>
<td>7</td>
<td>10</td>
<td>$10^{23}$</td>
</tr>
<tr>
<td>x264</td>
<td>Video Encoder</td>
<td>8</td>
<td>13</td>
<td>0</td>
<td>$10^{27}$</td>
</tr>
</tbody>
</table>
2. Experiment: Performance Prediction

RQ: Which combination of sampling approaches achieve the highest prediction accuracy?

<table>
<thead>
<tr>
<th>System</th>
<th>Domain</th>
<th># Binary Opt.</th>
<th># Numeric Opt.</th>
<th># Constraints</th>
<th># Configs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dune MGS</td>
<td>Multi-Grid Solver</td>
<td>8</td>
<td>3</td>
<td>20</td>
<td>2 304</td>
</tr>
<tr>
<td>HIPA cc</td>
<td>Image Processing</td>
<td>31</td>
<td>2</td>
<td>416</td>
<td>13 485</td>
</tr>
<tr>
<td>HSMGP</td>
<td>Multi-Grid Solver</td>
<td>11</td>
<td>3</td>
<td>45</td>
<td>3 456</td>
</tr>
<tr>
<td>JavaGC</td>
<td>Runtime Env.</td>
<td>12</td>
<td>22</td>
<td>4</td>
<td>10^{31}</td>
</tr>
<tr>
<td>x264</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0^{27}</td>
</tr>
</tbody>
</table>

Average prediction accuracy (PW+PBD): **86.8%**!
Experimental designs are better than random sampling.
3. Experiment: Accuracy vs. Complexity

• **RQ:** Is it necessary to learn accurate but complex models?

It depends on the use case. Simple models help to identify the most relevant influences.
4. Experiment: Validation of Domain Knowledge

• **RQ:** Are we able to validate domain knowledge using a model?

   ![Diagram showing the process of validating domain knowledge]

   - **Domain Expert** → **Sampling** → **Learning** → **Theoretical Knowledge** → **Perf.-influence model** → **Validate domain knowledge**
Findings and Future Work

Evaluation: Performance Prediction

Validation of Domain Knowledge

Accuracy vs. Complexity

- RQ: Is it necessary to learn accurate but complex models?

It depends on the use case. Simple models help to identify the most relevant influences.

Thank You!

https://github.com/nsiegmun/SPLConqueror
http://fosd.de/SPLConqueror/