Automatic Selection of Compiler Options Using Non-parametric Inferential Statistics

Masayo Haneda
Peter M.W. Knijnenburg
Harry A.G. Wijshoff
LIACS, Leiden University
Motivation

- An optimal compiler optimization setting can be obtained by considering the interaction between applications, architectures, and compiler optimizations.
- Profiling is the best way to understand this interaction.
- However, a huge number of optimization settings exists.
- Find the optimal configuration with limited amount of profiling.
Outline of Our Approach

• Collect profiling data with an appropriate experimental setting
  ○ Orthogonal Arrays, which are well known in the Design of Experiments, are used for this purpose

• Apply Inferential Statistics to the profiling data to detect effective compiler options
  ○ Mann-Whitney test of non-parametric inferential statistics is employed
Orthogonal Arrays

- Orthogonal arrays (OAs) are well chosen fractional factorial designs.
- An OA is expressed as a $N \times k$ matrix of 0s and 1s.
- The columns are interpreted as factors (compiler options).
- Each row of an array defines a compiler setting.
Orthogonal Arrays (cont’d)

- An OA has a property that two arbitrary columns contain the patterns
  00  01  10  11
equally often.

- According to this property,
  - each compiler option is turned on and off equally often.
  - When we drop columns of an OA, the array is still an orthogonal array.

- An OA exists when the number of rows \( N \) is a multiple of 4
Example

\[
\begin{pmatrix}
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0
\end{pmatrix}
\]

Interpreted as Compiler Settings

<table>
<thead>
<tr>
<th>Run</th>
<th>O₁</th>
<th>O₂</th>
<th>O₃</th>
<th>O₄</th>
<th>O₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run1</td>
<td>off</td>
<td>off</td>
<td>off</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>Run2</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>Run3</td>
<td>off</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>Run4</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>Run5</td>
<td>on</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>Run6</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>Run7</td>
<td>off</td>
<td>on</td>
<td>on</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>Run8</td>
<td>on</td>
<td>on</td>
<td>on</td>
<td>off</td>
<td>off</td>
</tr>
</tbody>
</table>
Inferential Statistics

- Inferential statistics is used to predict whether a factor of an experiment has a significant effect.

- Inferential statistics uses the logic of null hypothesis and a test statistic.
Null Hypothesis

* Null hypothesis denies the effect of a factor in an experiment.
* When a compiler option A is applied on an application B:
  OCompiler option A is not effective to optimize application B.
* When the null hypothesis is negated, we can say the compiler option A is effective to optimize application B.
* Test statistic is computed from the experimental data to evaluate likelihood of the null hypothesis.
Mann-Whitney Test

• A well known test in non-parametric inferential statistics.

• Uses ranked order from experimental data to be applied the experimental data with a few assumption

• The experimental data is divided into 2 parts, experimental group and control group according to the null hypothesis.
Mann-Whitney Test (Cont’d)

• Null hypothesis:
  ○ Compiler option A is not effective to optimize application B.

• Experimental group contains the compiler settings using compiler option A

• Control group contains the compiler settings which do not use compiler option A
Test Statistic $z$

- Test statistic $z$ represents how many units of standard deviation are in the distance between mean value and observed value.
- $Z$ for observed value $k$ is computed as follows.

$$z = \frac{k - \mu}{\sigma}$$

- $\sigma$: Standard deviation of experimental data
- $\mu$: Mean of experimental data

- $Z$ is converted to the probability that the observed value occurs.
Z to Probability

- Ranked data has a normal distribution.
- When the distribution is normal, it is possible to compute the cumulative probability to observe the value within $|z|$.
- When $t=|z|$, $P(t)$ expresses the cumulative percentage of data values out of $|z|$.

\[ P(t) = (1 - 2 \cdot \int_0^{|z|} \frac{1}{\sigma \cdot \sqrt{2\pi}} e^{-\frac{1}{2}z^2} \, dz) \cdot 100\% \]

- In the theory of statistics, a z-value satisfying $P(t)<5\%$ is considered significant.
Z to Probability

![Normal Distribution with Z Values]
Example

- Find important options from the 10 compiler options

<table>
<thead>
<tr>
<th>Run1</th>
<th>O₁</th>
<th>O₂</th>
<th>O₃</th>
<th>O₄</th>
<th>O₅</th>
<th>O₆</th>
<th>O₇</th>
<th>O₈</th>
<th>O₉</th>
<th>O₁₀</th>
<th>Time</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run2</td>
<td>1 0 1 0 0 0 1 1 1 0</td>
<td>25</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run3</td>
<td>1 1 0 1 0 0 0 1 1 1</td>
<td>15</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run4</td>
<td>0 1 1 0 1 0 0 0 1 1</td>
<td>17</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run5</td>
<td>1 0 1 1 0 1 0 0 0 1</td>
<td>18</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run6</td>
<td>1 1 0 1 1 0 1 0 0 0</td>
<td>14</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run7</td>
<td>1 1 1 0 1 1 0 1 0 0</td>
<td>23</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run8</td>
<td>0 1 1 1 0 1 1 0 1 0</td>
<td>13</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run9</td>
<td>0 0 1 1 1 0 1 1 0 1</td>
<td>19</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run10</td>
<td>0 0 0 1 1 1 0 1 1 0</td>
<td>22</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run11</td>
<td>1 0 0 0 1 1 1 0 1 1</td>
<td>21</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run12</td>
<td>0 1 0 0 0 1 1 1 0 1</td>
<td>16</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example (Cont’d)

● Is $O_2$ an important option?

<table>
<thead>
<tr>
<th></th>
<th>- $O_2$ -</th>
<th>Time</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run1</td>
<td>0</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>Run2</td>
<td>0</td>
<td>25</td>
<td>12</td>
</tr>
<tr>
<td>Run3</td>
<td>1</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>Run4</td>
<td>1</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>Run5</td>
<td>0</td>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>Run6</td>
<td>1</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>Run7</td>
<td>1</td>
<td>23</td>
<td>11</td>
</tr>
<tr>
<td>Run8</td>
<td>1</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>Run9</td>
<td>0</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>Run10</td>
<td>0</td>
<td>22</td>
<td>10</td>
</tr>
<tr>
<td>Run11</td>
<td>0</td>
<td>21</td>
<td>9</td>
</tr>
<tr>
<td>Run12</td>
<td>1</td>
<td>16</td>
<td>4</td>
</tr>
</tbody>
</table>

Null hypothesis: $O_2$ is not an important compiler option

<table>
<thead>
<tr>
<th>Experimental Group($O_2=1$)</th>
<th>Control Group($O_2=0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (Run3)</td>
<td>8 (Run1)</td>
</tr>
<tr>
<td>5 (Run4)</td>
<td>12 (Run2)</td>
</tr>
<tr>
<td>2 (Run6)</td>
<td>6 (Run5)</td>
</tr>
<tr>
<td>11 (Run7)</td>
<td>7 (Run9)</td>
</tr>
<tr>
<td>1 (Run8)</td>
<td>10 (Run10)</td>
</tr>
<tr>
<td>4 (Run12)</td>
<td>9 (Run11)</td>
</tr>
</tbody>
</table>

Divide into 2 groups

Compute test statistic $z$
Example

- Compute test statistic $z$ for $O_2$

<table>
<thead>
<tr>
<th>Experimental Group ($O_2=1$)</th>
<th>Control Group2 ($O_2=0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (Run3)</td>
<td>8 (Run1)</td>
</tr>
<tr>
<td>5 (Run4)</td>
<td>12 (Run2)</td>
</tr>
<tr>
<td>2 (Run6)</td>
<td>6 (Run5)</td>
</tr>
<tr>
<td>11 (Run7)</td>
<td>7 (Run9)</td>
</tr>
<tr>
<td>1 (Run8)</td>
<td>10 (Run10)</td>
</tr>
<tr>
<td>4 (Run12)</td>
<td>9 (Run11)</td>
</tr>
</tbody>
</table>

$$\sigma = \sqrt{\frac{6 \cdot 6 \cdot (6+6+1)}{12}} = \sqrt{39}$$

$$\mu = \frac{6(6+6+1)}{2} = 39$$

$$k = 3 + 5 + 2 + 11 + 1 + 4 = 26$$

$$z = \frac{k - \mu}{\sigma} = -2.08$$

$$P|{-2.08}| < 5\%$$
Iterative Algorithm

- Choose an orthogonal array $A$ with as many columns as there are options.
- Repeat
  - Compile the application with each row from $A$ as compiler setting and execute the optimized application.
  - Compute test statistic $z$ for each compiler option
  - If the test statistic meets $P(\lvert z\rvert) < 5$
    - If $z$ is negative then the option is turned on.
    - If $z$ is positive then the option is turned off.
  - Remove the compiler options that have been selected from the factor list and drop the same number of columns from $A$.
- Until
  - All options are set, or
  - No option with a significant effect is detected anymore, or
  - The experimental data has not enough variation (low standard deviation) to apply the Mann-Whitney test meaningfully.
- Choose the compiler setting which has the best execution time in the last experiment.
Application to GCC

- Compiler version: 3.3.1
- Number of options: 45 options
  - We arrange them into 23 factors according to their dependency.
- Architecture: Pentium 4 at 2.8GHz
- Applications: 10 programs from the SPEC 2000 benchmark suite with the train data set.
- Measurement: Clockticks observed by using VTune (a tool to analyze performance of applications on Intel architectures)
Case Study (181.mcf)

- We illustrate the performance of our algorithm with the results per iteration for 181.mcf.
- 23 options to be configured
- A 24x23 orthogonal array is chosen to design the experiment
- 181.mcf is compiled with 24 settings, and executed.
24 data points (1\textsuperscript{st} Iteration)
1st Iteration (P-value)

Options

- Significant Effect (5%)
- Negative Effect
- Positive Effect

| O1 | O2 | O3 | O4 | O5 | O6 | O7 | O8 | O9 | O10 | O11 | O12 | O13 | O14 | O15 | O16 | O17 | O18 | O19 | O20 | O21 | O22 | O23 |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
24 data points (2nd Iteration)
2nd Iteration (P-value)

Options

<table>
<thead>
<tr>
<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>
<th>O_9</th>
<th>O_{10}</th>
<th>O_{11}</th>
<th>O_{12}</th>
<th>O_{13}</th>
<th>O_{14}</th>
<th>O_{15}</th>
<th>O_{16}</th>
<th>O_{17}</th>
<th>O_{18}</th>
<th>O_{19}</th>
<th>O_{20}</th>
<th>O_{21}</th>
<th>O_{22}</th>
<th>O_{23}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
24 data points (3rd Iteration)
Results (SPEC2000 CINT)

Benchmarks

Improvement

0% 10% 20% 30% 40% 50% 60% 70%

gzip vpr mcf parser gap vortex bzip2

OA24 OA48 OA100 O3
Result (SPEC2000 CFP)

Bar chart showing improvement percentages for benchmarks 'wupwise', 'swim', and 'art' with different configurations.
Summary

- We have introduced the concept of Design of Experiments to determine compiler settings.
- We have employed the Mann-Whitney test to examine the significance of the effect of compiler options.
- Select important compiler options iteratively and obtain one optimization setting for each application.
- All settings which we obtained for each benchmark outperform O3 which is the standard setting of gcc.
Sample Size

- Power analysis provides the adequate sample size according to the variance of sample data and strength of reliability.

- Sample size is given by using Z-values of the threshold for the Mann-Whitney test ($Z_\alpha$) and the reliability of the test ($Z_\beta$)

\[ M = \left[ 5 \cdot (Z_\alpha + Z_\beta)^2 \right] \]

- $\alpha$ is 5%, and for $\beta$ we use 60%, 85%, and 99.5% which are OA size 24, 48, and 100 respectively