Including Improvement of the Execution Time in a Software Architecture of Libraries with Self-Optimisation

Luis-Pedro García
Javier Cuenca
Domingo Giménez

University of Murcia
SPAIN
Outline

- Introduction
- Self-Optimised Linear Algebra Routine Samples
- Experimental Results
- Conclusions
Outline

- Introduction
- Self-Optimised Linear Algebra Routine Samples
- Experimental Results
- Conclusions
Our goal: to obtain linear algebra parallel routines with auto-optimization capacity.

The approach: model the execution time of the routine to tune, taking advantage of the natural hierarchy existing in linear algebra programs.

The basic idea is to start from lower level routines (multiplication, addition, etc.) To model the higher level ones (Strassen multiplication, parallel multiplication, LU, QR, Cholesky, etc).

In this talk:
- A remodelling stage is proposed if the information at one level is not accurate enough.
- This new model will be built using polynomial regression.
Introduction

Tuned_Model

Tuning

Tested_Model

Optimum_AP

Selection of Optimum_AP

Optimum_AP

Execution of the LAR

Runtime system information

Tested_Model

Tuning Tested_Model

Testing Theoretical_Model

SOLAR_manager

Modelling the LAR theoretically

Implementation of the manager

Theoretical_Model

LAR

Installation

Design

Execution

Values of $n$

Values of AP

Model_Testing_Value

$n_k$
Introduction

Theoretical and experimental study of the algorithm.

An analytical model of the execution time

\[ T(n) = f(n, AP, SP) = 2n^3 k_3 \ (\text{dgemm}) \]

In linear algebra parallel routines, typical \( SP \) are:

\[ k_1, k_2, k_3, t_s \text{ and } t_w \]

…and \( AP \) are:

\[ b, p = r \times c \text{ and the basic library} \]
Introduction

- Theoretical and experimental study of the algorithm.
- An analytical model of the execution time:
  \[ T(n) = f(n, AP, SP) = 2n^3 k_3 \ (dgemm) \]
- In linear algebra parallel routines, typical \( AP \) are:
  - \( b, p = r \times c \) and the basic library
- …and \( SP \) are:
  - \( k_1, k_2, k_3, t_s \) and \( t_w \)
Introduction

- Tuned_Model
- Tuning
- Tested_Model
- Optimum_AP
- Execution of the LAR
- Optimum_AP
- Selection of Optimum_AP
- SOLAR_manager
- Theory
- Modelling the LAR theoretically
- Implementation of the manager
- Theoretical_Model
- Model_Testing_Varlues
- Values of n
- Values of AP
- n_0
- Tuned_Model
- Tuning Tested_Model
- Runtime system information
- Information
- NWS
Introduction

Tuned_Model

Modelling the LAR theoretically

Theoretical_Model

Implementation of the manager

SOLAR_manager

Execution of the LAR

Optimum_AP

Selection of Optimum_AP

Values of n

Values of AP

Model_Testing_Value

n

Tuned_Model

Testing Theoretical_Model

Tested_Model

Runtime system information

NWS
Introduction

LAR
Modelling the LAR theoretically

Theoretical_Model

Implementation of the manager

SOLAR_manager

Model_Testing Values

Values of n
Values of AP

Execution of the LAR
Optimum_AP
Selection of Optimum_AP

Tuned_Model

Testing Theoretical_Model

Tested_Model

Tuning Tested_Model
Runtime system information

NWS
Introduction

Tuned_Model

Testing Theoretical_Model

Tested_Model

LAR

Modelling the LAR theoretically

Theoretical_Model

Implementation of the manager

SOLAR_manager

Execution of the LAR

Optimum_AP

Selection of Optimum_AP

Tuned_Model

Tuning Tested_Model

Runtime system information

NWS

Values of n

Values of AP

Model_Testing_Value_s

ICSOFT 2007
Testing the model:
Remodelling de Linear Algebra Routine \((L\text{AR})\)

Designing a polynomial scheme from the original model for different combinations of \(n\) and \(AP\):

\[
T(n, AP) = a_0 n^3 / p + a_1 n^3 p + a_2 n^3 + a_3 n^2 / p + a_4 n^2 p + a_5 n^2 + \ldots
\]

The coefficients \(a_0, a_1, a_2, \ldots\) must be calculated.
Introduction

LAR

Modelling the LAR theoretically

Theoretical_Model

Implementation of the manager

SOLAR_manager

Model_Testing_Value

Values of n

Values of AP

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

Tested_Model

Testing Theoretical_Model

NWS
Introduction

LAR
- Modelling the LAR theoretically
  - Theoretical_Model
    - Implementation of the manager
      - SOLAR_manager
        - Testing Theoretical_Model
          - Tested_Model
            - Tuned_Model
              - Tuning Tested_Model
                - Runtime system information
                  - NWS
  - Selection of Optimum_AP
    - Optimum_AP
      - Model_Testing_Value
        - Values of n
        - Values of AP
          - $n_k$
Introduction

Tuned_Model
Tuning Tested_Model
Optimum_AP
Selection of Optimum_AP
Execution of the LAR

LAR
Modelling the LAR theoretically
Theoretical_Model
Implementation of the manager
SOLAR_manager

Model_Testing_Values
Values of n
Values of AP

Optimum_AP
Values of AP
n_k

Tuned_Model
Tuning Tested_Model
Runtime system information
Information
NWS

Testing Theoretical_Model
Tested_Model

Modelling the LAR theoretically
Implementation of the manager
SOLAR_manager

Execution of the LAR
Optimum_AP
Selection of Optimum_AP

Introduction

Tuned_Model

Testing Theoretical_Model

Tested_Model

LAR

Modelling the LAR theoretically

Theoretical_Model

Implementation of the manager

SOLAR_manager

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS

Model_Testing_Value

Values of $n$

Values of AP

Introduction

Optimum_AP

Selection of Optimum_AP

Execution of the LAR

Optimum_AP

$n_A$

Tuned_Model

Tuning Tested_Model

Runtime system information

Information

NWS
Introduction

LAR
Modelling the LAR theoretically
Theoretical_Model
Implementation of the manager
SOLAR_manager
Testing Theoretical_Model
Tested_Model
Execution of the LAR
Optimum_AP
Selection of Optimum_AP
\( r_s \)
Tuned_Model
Tuning Tested_Model

Model_Testing_Values
Values of \( n \)
Values of AP
Runtime system information
Information
NWS
Remodelling de Linear Algebra Routine (LAR)

Designing a polynomial scheme from the original model for different combinations of $n$ and $AP$:

$$T(n, AP) = \frac{a_0 n^3}{p} + a_1 n^3 p + a_2 n^3 + \frac{a_3 n^2}{p} + a_4 n^2 p + a_5 n^2 + \ldots$$

The coefficients $a_0, a_1, a_2, \ldots$ must be calculated.
Introduction

In order to determine these coefficients, four different methods are proposed:

- **FI-ME**: Fixed Minimal Executions
- **VA-ME**: Variable Minimal Executions
- **FI-LS**: Fixed Least Square
- **VA-LS**: Variable Least Square

OCULTA

ICSOFT 2007
Outline

- Introduction
- Self-Optimised Linear Algebra Routine Samples
- Experimental Results
- Conclusions
Self-Optimised LAR

- Strassen Matrix-Matrix multiplication

\[ T = 7^l t_{mult} \left( \frac{n}{2^l} \right) + 18 \sum_{i=1}^{l} 7^{i-1} t_{add} \left( \frac{n}{2^i} \right) \]

- \( t_{mult}(n/2^l) \): Theoretical execution time for matrix multiplication. BLAS3 function DGEMM

- \( t_{add}(n/2^i) \): Theoretical execution time for matrix addition. BLAS1 function DAXPY
Experimental Results: Strassen

- **Systems:**
  - Xeon: Linux Intel Xeon 3.0 GHz workstation
  - Alpha: Unix HP-Alpha 1.0 GHz workstation

- **Models for DGEMM and DAXPY**
  - **DGEMM:** Third order polynomial (20 samples)
    - $n_{\text{min}} = 500$, $n_{\text{max}} = 10000$, $n_{\text{inc}} = 500$
  - **DAXPY:** Sixth order polynomial (31 samples)
    - $n_{\text{min}} = 64$, $n_{\text{max}} = 2000$, $n_{\text{inc}} = 64$
## Experimental Results: Strassen

- **Testing de Model in Xeon.**
  (Time in seconds)

<table>
<thead>
<tr>
<th>$n$</th>
<th>$l$</th>
<th><strong>Mod.</strong></th>
<th><strong>Exp.</strong></th>
<th><strong>Dev. (%)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>3072</td>
<td>1</td>
<td>11.75</td>
<td>12.86</td>
<td>8.58</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>13.90</td>
<td>13.63</td>
<td>1.99</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>37.04</td>
<td>15.76</td>
<td>135.06</td>
</tr>
<tr>
<td>4096</td>
<td>1</td>
<td>27.21</td>
<td>29.71</td>
<td>8.41</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>28.59</td>
<td>30.10</td>
<td>5.02</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>48.76</td>
<td>33.34</td>
<td>46.26</td>
</tr>
<tr>
<td>5120</td>
<td>1</td>
<td>53.14</td>
<td>56.83</td>
<td>6.51</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>53.53</td>
<td>56.43</td>
<td>5.13</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>71.08</td>
<td>60.19</td>
<td>18.09</td>
</tr>
<tr>
<td>6144</td>
<td>1</td>
<td>96.48</td>
<td>96.32</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>95.39</td>
<td>93.69</td>
<td>1.82</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>110.40</td>
<td>98.39</td>
<td>12.21</td>
</tr>
</tbody>
</table>

- **Testing de Model in Alpha.**
  (Time in seconds)

<table>
<thead>
<tr>
<th>$n$</th>
<th>$l$</th>
<th><strong>Mod.</strong></th>
<th><strong>Exp.</strong></th>
<th><strong>Dev. (%)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>3072</td>
<td>1</td>
<td>29.96</td>
<td>29.70</td>
<td>0.89</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>28.54</td>
<td>27.82</td>
<td>2.57</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>17.55</td>
<td>27.61</td>
<td>36.46</td>
</tr>
<tr>
<td>4096</td>
<td>1</td>
<td>69.85</td>
<td>70.85</td>
<td>1.43</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>66.04</td>
<td>64.55</td>
<td>2.30</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>57.82</td>
<td>62.56</td>
<td>7.58</td>
</tr>
<tr>
<td>5120</td>
<td>1</td>
<td>135.03</td>
<td>134.67</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>125.76</td>
<td>123.38</td>
<td>1.92</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>118.12</td>
<td>118.45</td>
<td>0.28</td>
</tr>
<tr>
<td>6144</td>
<td>1</td>
<td>229.79</td>
<td>232.27</td>
<td>1.07</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>211.10</td>
<td>210.88</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>201.15</td>
<td>199.33</td>
<td>0.92</td>
</tr>
</tbody>
</table>
The optimal value of $AP$ vary for different systems and problem sizes.

In Xeon and for $n = 5120$ the model make a wrong prediction, but the execution time is only 0.71% higher.

However, in Xeon, the deviation ranged from 0.17% to 135.06%:

*IT IS NECESSARY TO BUILD AN IMPROVED MODEL*
The scheme consists of defining a set of third grade polynomial functions from the theoretical model:

\[ T(n, l) = 2 \times 7^l \left( \frac{n}{2^l} \right)^3 M(l) + \frac{18}{4} n^2 A(l) \sum_{i=1}^{l} \left( \frac{7}{4} \right)^{i-1} \]

- \( M(l) \) and \( A(l) \) must be calculated.
- For each \( l \), \( n \) varies and the values of \( M(l) \) and \( A(l) \) are obtained by least squares.
Remodelling Strassen

- The scheme consists of defining a set of third grade polynomial functions from the theoretical model:

\[
\frac{\sum_{l} \text{M}(l)}{\prod_{l} \text{A}(l)}
\]

- \(\text{M}(l)\) and \(\text{A}(l)\) must be calculated.
- For each \(l\), \(n\) varies and the values of \(\text{M}(l)\) and \(\text{A}(l)\) are obtained by least squares.

<table>
<thead>
<tr>
<th>(l)</th>
<th>(\text{M}(l))</th>
<th>(\text{A}(l))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2.22 \times 10^{-10})</td>
<td>(3.89 \times 10^{-08})</td>
</tr>
<tr>
<td>2</td>
<td>(2.24 \times 10^{-10})</td>
<td>(3.03 \times 10^{-08})</td>
</tr>
<tr>
<td>3</td>
<td>(1.99 \times 10^{-10})</td>
<td>(3.03 \times 10^{-08})</td>
</tr>
<tr>
<td>4</td>
<td>(3.48 \times 10^{-10})</td>
<td>(1.53 \times 10^{-08})</td>
</tr>
</tbody>
</table>
Remodelling Strassen

- Now the set of values for $M(l)$ and $A(l)$ can be approximated by a polynomial in $l$ and thus we have a single model for any combination of $n$ and $l$.

- $M(l)$ is approximated by a second grade polynomial

\[ M(l) = m_0 + m_1 l + m_2 l^2 \]

- $A(l)$ is approximated by a first grade polynomial

\[ A(l) = a_0 + a_1 l \]
Now the set of values for $M(l)$ and $A(l)$ can be approximated by a polynomial in $l$, and thus we have a single model for any combination of $n$ and $l$.

- $M(l)$ is approximated by a second grade polynomial
  \[ M(l) = m_0 + m_1 l + m_2 l^2 \]

- $A(l)$ is approximated by a first grade polynomial
  \[ A(l) = a_0 + a_1 l \]
## Remodelling Strassen

<table>
<thead>
<tr>
<th>$n$</th>
<th>$l$</th>
<th>Mod.</th>
<th>Exp.</th>
<th>Dev. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2688</td>
<td>1</td>
<td>7.87</td>
<td>8.80</td>
<td>11.92</td>
</tr>
<tr>
<td>2688</td>
<td>2</td>
<td>8.40</td>
<td>9.67</td>
<td>15.23</td>
</tr>
<tr>
<td>2688</td>
<td>3</td>
<td>10.28</td>
<td>10.52</td>
<td>2.38</td>
</tr>
<tr>
<td>3200</td>
<td>1</td>
<td>13.02</td>
<td>14.51</td>
<td>11.92</td>
</tr>
<tr>
<td>3200</td>
<td>2</td>
<td>13.56</td>
<td>15.51</td>
<td>14.38</td>
</tr>
<tr>
<td>3200</td>
<td>3</td>
<td>16.00</td>
<td>16.30</td>
<td>1.87</td>
</tr>
<tr>
<td>5120</td>
<td>1</td>
<td>56.80</td>
<td>56.71</td>
<td>0.17</td>
</tr>
<tr>
<td>5120</td>
<td>2</td>
<td>56.44</td>
<td>57.01</td>
<td>1.00</td>
</tr>
<tr>
<td>5120</td>
<td>3</td>
<td>60.04</td>
<td>55.09</td>
<td>8.25</td>
</tr>
<tr>
<td>5632</td>
<td>1</td>
<td>75.78</td>
<td>74.92</td>
<td>1.12</td>
</tr>
<tr>
<td>5632</td>
<td>2</td>
<td>73.50</td>
<td>74.56</td>
<td>1.45</td>
</tr>
<tr>
<td>5632</td>
<td>3</td>
<td>71.70</td>
<td>70.97</td>
<td>1.03</td>
</tr>
</tbody>
</table>

---

ICSOFT 2007
In Xeon and for $n = 5120$ the model makes a wrong prediction, but the execution time is only 3.49% higher.

Now, with remodelling, the deviation is smaller and ranged from 0.17% to 15.23%.
Outline

- Introduction
- Self-Optimised Linear Algebra Routine Samples
- Experimental Results
- Conclusions
Conclusions

- The use of modelling techniques can contribute to reduce the execution time of the routines.

- The modelling time must be small:
  - Reduce the number of samples.
  - Use small problem sizes for modelling.

- The method has been applied successfully to the Strassen Matrix-Matrix multiplication and can be applied to other linear algebra routines.