An Architectural Template for Parallel Loops and Sections

Symposium on Software Performance 2018

Markus Frank & Alireza Hakamian
The Problem with Palladio and Parallelism

John
Data Analyst

Me
Software Performance Engineer

Palladio Simulator

Manual Modelling Overhead

Inaccurate Predictions

Performance Prediction Results

Off by 63 %


Idea and Process

Common parallelisation patterns as easy to use architectural templates

Pattern Identification (Code) → Pattern Modelling (Palladio) → Architectural Template Creation → Accuracy Improvement
Pattern Identification

Pattern Identification (Code)
Pattern Modelling (Palladio)
Architectural Template Creation
Accuracy Improvement
Parallel Loop

Code example

```java
for (int i = 0; i < matrixA.getWidth(); i++) {
    for (int k = 0; k < matrixB.getHeight(); k++) {
        for (int j = 0; j < matrixA.getHeight(); j++) {
            result[i][j] += matrixA[i][k] * matrixB[k][j];
        }
    }
}
```

John
Data Analyst
Pattern Modelling

Pattern Identification (Code)

Pattern Modelling (Palladio)

Architectural Template Creation

Accuracy Improvement
Parallel Loop
SEFF Example (Single Core)

rep = matrixASizeM.VALUE * matrixASizeN.VALUE * matrixBSizeJ.VALUE

Resource Demands
0.00000069 <CPU>

<< InternalAction >>
calculation

Resource Demands
0.00000069 <CPU>
Parallel Loop
SEFF Example (2 Threads)

Software Model

<< Fork >>
Forked Behaviours
<< Synchronisation Point >>

<< InternalAction >>
calculationA

Resource Demands
0.00000069 * matrixASizeM.VALUE * matrixASizeN.VALUE * matrixBSizeJ.VALUE / 2 <CPU>

<< InternalAction >>
calculationB

Resource Demands
0.00000069 * matrixASizeM.VALUE * matrixASizeN.VALUE * matrixBSizeJ.VALUE / 2 <CPU>

Resource Demand
Repetitions
Parallelisation
Parallel Loop

SEFF Example (18 Threads)

Manual modelling overhead increase with each thread
But, what we want to have...

\[
\text{rep} = \text{matrixASizeM.VALUE} \times \text{matrixASizeN.VALUE} \times \text{matrixBSizeJ.VALUE}
\]

\[
\text{threadPooleSize} = \text{threadNumber}
\]

Efficiency

Software Model

Resource Demand

Repetitions

Parallelisation

Resource Demand:
- Repetitions: 10

Efficiency: M. Frank - Symposium on Software Performance 2018 - An Architectural Template for Parallel Loops
Architectural Templates

- Pattern Identification (Code)
- Pattern Modelling (Palladio)
- **Architectural Template Creation**
- Accuracy Improvement
3-Steps for AT

1. Define Stereotype

2. OVT-o

3. Add To Catalog

"Architectural Templates allow software architects to apply reusable patterns to their Palladio models."

M. Frank - Symposium on Software Performance 2018 - An Architectural Template for Parallel Loops

Accuracy Improvement

Pattern Identification (Code)

Pattern Modelling (Palladio)

Architectural Template Creation

Accuracy Improvement
Adding Overhead to Loop AT

```
<< loopAction >>
@Parallel
rep = matrixASizeM.VALUE * matrixASizeN.VALUE * matrixBSIZEJ.VALUE

threadPoolSize = threadNumber
overhead = threadPoolSize * 0.0000079 <CPU>

<< InternalAction >>
calculation
ResourceDemands
0.00000069 <CPU>
```

Each thread needs additional resources to span and synchronize.

Additional overhead due to communication or waiting condition may occur.
Adding Overhead to Model

<< Fork >>

ForkedBehaviours

<< Synchronisation Point >>

<< InternalAction >>
calculationA

ResourceDemands:
0.00000069 * matrixASizeM.VALUE * 
matrixASizeN.VALUE * 
matrixBSizeJ.VALUE / 2 <CPU>

<< InternalAction >>
calculationB

ResourceDemands:
0.00000069 * matrixASizeM.VALUE * 
matrixASizeN.VALUE * 
matrixBSizeJ.VALUE / 2 <CPU>

<< InternalAction >>
overhead

ResourceDemands:
overhead <CPU>
Limitations & Future Work
Limitations & Future Work

**Limitations**

- Equal workload
- Only OpenMP-like
- Estimation of overhead function

**Future Work**

- MPI and ACTORS
- Reference curves for different resource demands
- Abstraction to architectural level
- Include additional properties in Simulations
The Problem with Palladio and Parallelism

John
Data Analyst

Me
Software Performance Engineer

Palladio Simulator

Performance Model

Manual Modelling Overhead

Inaccurate Predictions

Performance Prediction Results

Off by 63%
Thank you!

Markus Frank

E-Mail markus.frank@informatik.uni-stuttgart.de
Telefon +49 (0) 711 685-88272
www.iste.uni-stuttgart.de/rss.html

Universität Stuttgart
Reliable Software Systems
Universitätsstraße 38, 70569 Stuttgart, Germany