

# A Cache Simulation Environment for OpenMP

---

Jie Tao, Thomas Brandes, and Michael Gerndt

Lehrstuhl für Rechnerarchitektur und Rechnerorganisation (LRR)  
Technische Universität München

Fraunhofer-Institute for Algorithms and Scientific Computing (SCAI)



# Motivation

---

## □ EP-Cache project

- ◆ Goal: cache locality optimization of realistic Fortran OpenMP applications on SMPs
- ◆ Infrastructure
  - ◆ Hardware monitors: information about cache access behavior
  - ◆ Monitoring control component: address transformation
  - ◆ Monitoring request interface: data aggregation
  - ◆ Performance tools: visualization, automatic performance analysis

## □ Simulation environment



# Outline

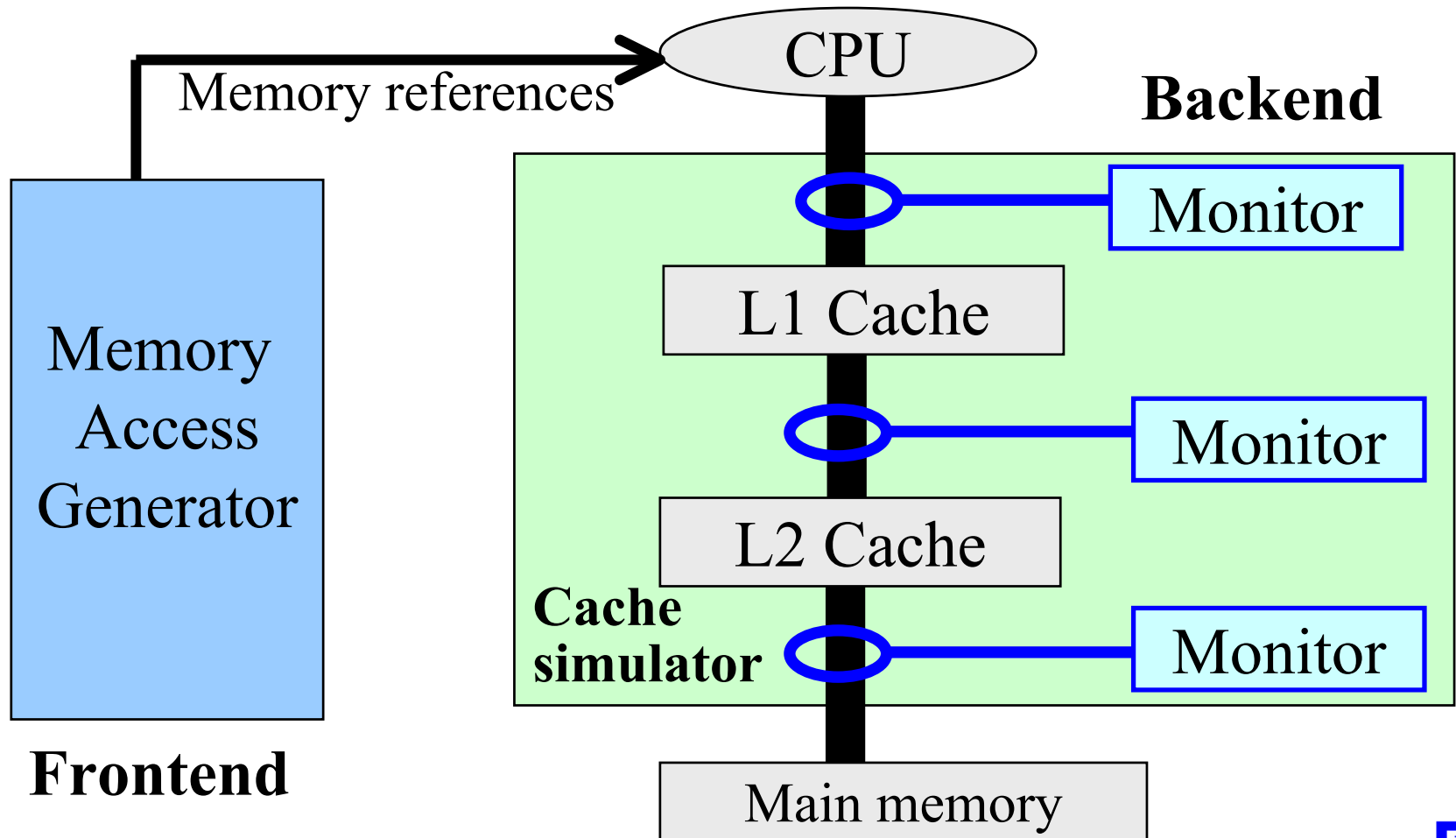
---

- What has to be simulated
- Basis platform
- OpenMP simulation
- Initial experimental results
- General OpenMP simulator
- Conclusions



# What has to be simulated

---



# What has to be simulated

---

- Parallel execution of OpenMP programs
  - ◆ Posix Threads to user-level threads
  - ◆ Thread scheduling
  
- Cache memory
  - ◆ Arbitrary level of caches
  - ◆ Various organisation: cache size, cache line size, associativity, latency, replacement
  - ◆ Different cache coherence protocols
  
- Hardware Monitor



# Basis Platform

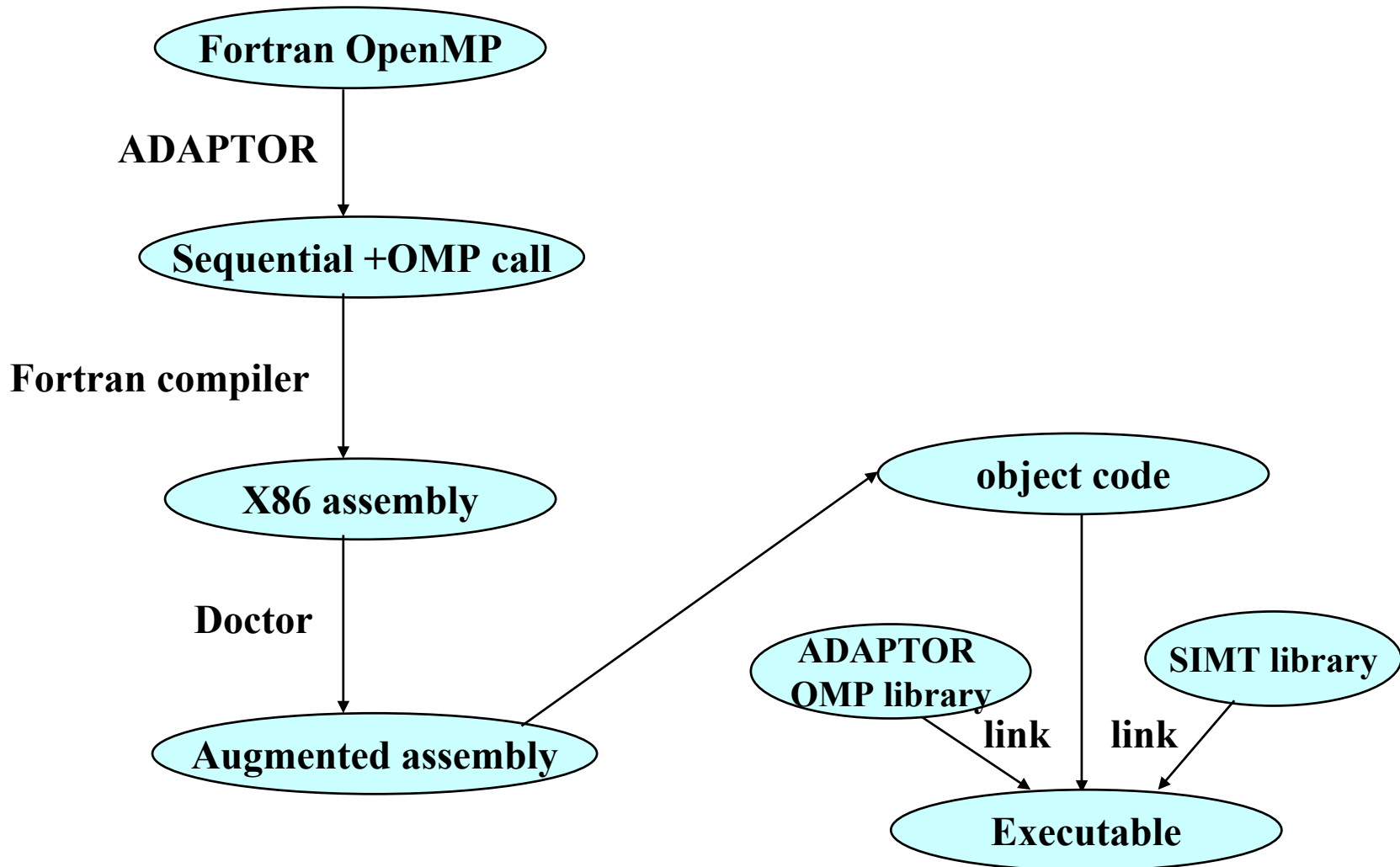
---

- SIMT: a multiprocessor simulator
  - ◆ Models NUMA machines, shared memory execution
  - ◆ Backend: target architecture simulator
    - ◆ Caches and coherence
    - ◆ DSM and data allocation
    - ◆ Network modeling
    - ◆ Hardware monitors
  - ◆ Frontend: Augmint
    - ◆ Basic simulation infrastructure for shared memory multiprocessor
    - ◆ Memory reference generation: instrumentation in assembly code
    - ◆ Parallelism: SPMD model, ANL like m4 micros
  
- ADAPTOR: source-to-source OpenMP compiler



# Building an Executable

---

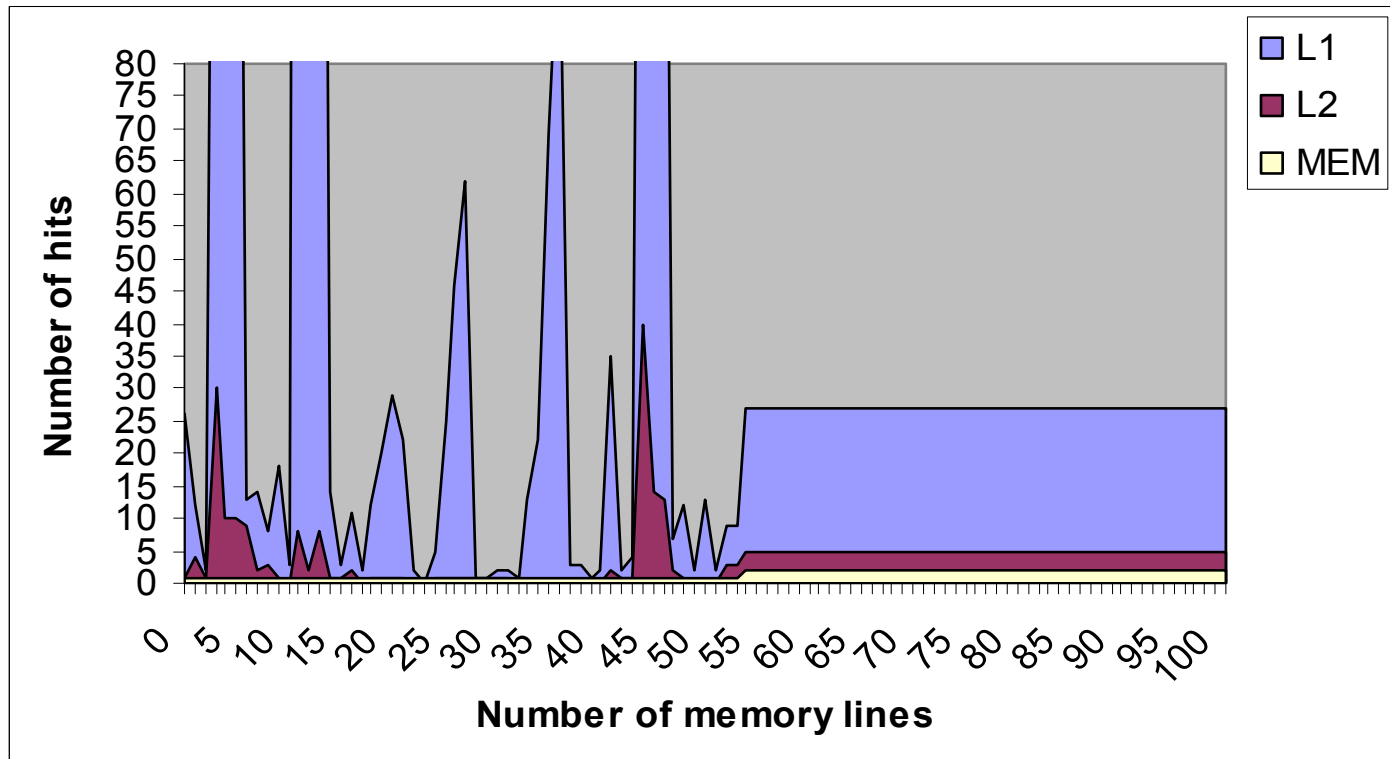


# OpenMP Simulation

---

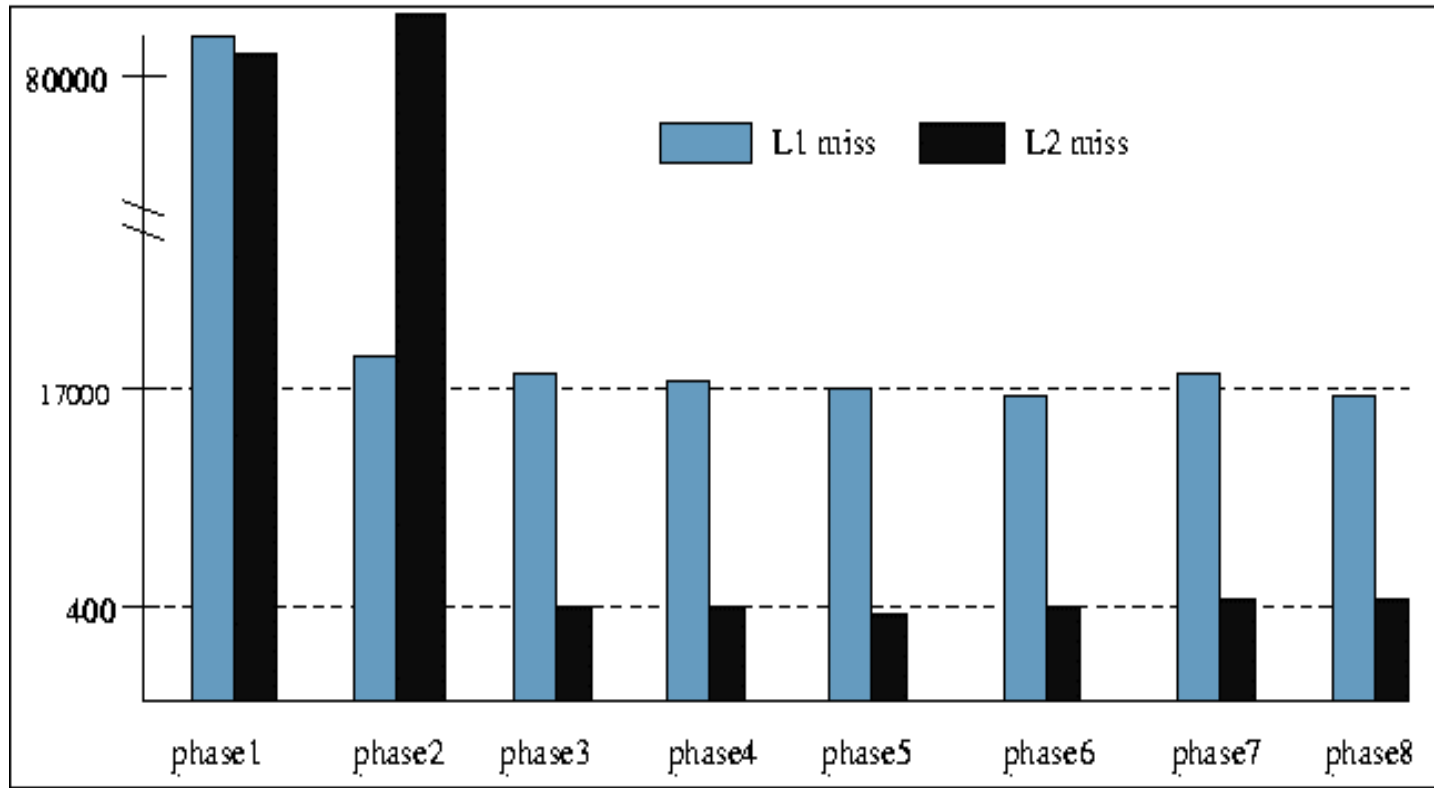
- ❑ Adaptive thread structure
  - ◆ create\_thread to SIMT CREATE
- ❑ Synchronization
  - ◆ Locks and barriers without deadlocks
- ❑ Scheduling
  - ◆ Supports only static scheduling
- ❑ Sequential and ordered regions
  - ◆ MASTER, CRITICAL, SINGLE, SECTION, ATOMIC
  - ◆ ORDERED

# Initial Experimental Results



Access histogram: jacobi on 8-nodes, 128\*128  
L1: 32KB, 2-way    L2: 512KB, 2-way





Temporal information with *heat*, working set  $256*256$



# Statistics on Cache Miss Source

---

<b>application</b>	<b>Variable</b>	<b>Miss rate</b>	<b>First refer</b>	<b>Replacement</b>	<b>Invalidation</b>
<b>matmul</b>	<b>A</b>	<b>62%</b>	<b>8193</b>	<b>110588</b>	<b>0</b>
	<b>X</b>	<b>26%</b>	<b>33</b>	<b>50962</b>	<b>0</b>
	<b>Y</b>	<b>10%</b>	<b>33</b>	<b>19701</b>	<b>0</b>
<b>jacobi</b>	<b>uold</b>	<b>21%</b>	<b>16436</b>	<b>59656</b>	<b>0</b>
	<b>u</b>	<b>36%</b>	<b>16384</b>	<b>113668</b>	<b>0</b>
	<b>f</b>	<b>41%</b>	<b>16424</b>	<b>130595</b>	<b>0</b>
<b>MG</b>	<b>u</b>	<b>32%</b>	<b>13814</b>	<b>748586</b>	<b>8608</b>
	<b>v</b>	<b>8%</b>	<b>14772</b>	<b>188406</b>	<b>47</b>
	<b>r</b>	<b>23%</b>	<b>10039</b>	<b>541581</b>	<b>5</b>



# General OpenMP Simulator

---

- Valgrind as Frontend
  - ◆ Memory debugger with cache profiler
  - ◆ Models PThreads
  - ◆ Runtime instrumentation
  
- SIMT cache simulator as backend
  - ◆ Pure interface to Valgrind
  
- General, independent
  - ◆ OpenMP compiler
  - ◆ Programming language (C, Fortran,...)
  - ◆ Programming model (using PThreads)



# Conclusions

---

## □ Contributions

- ◆ Cache simulator for OpenMP
  - ◆ Models the OpenMP execution on SMPs
  - ◆ Models caches and monitors
  - ◆ Provides comprehensive performance data
- ◆ First experimental results

## □ Future work

- ◆ Parallelization of the simulator (slowdown: 1000)
- ◆ Test with realistic applications
- ◆ Optimization with respect to cache locality

