STMBench7: A Benchmark for Software Transactional Memory

Rachid Guerraoui[1] Michał Kapałka[1] Jan Vitek[2]

[1]EPFL, Switzerland

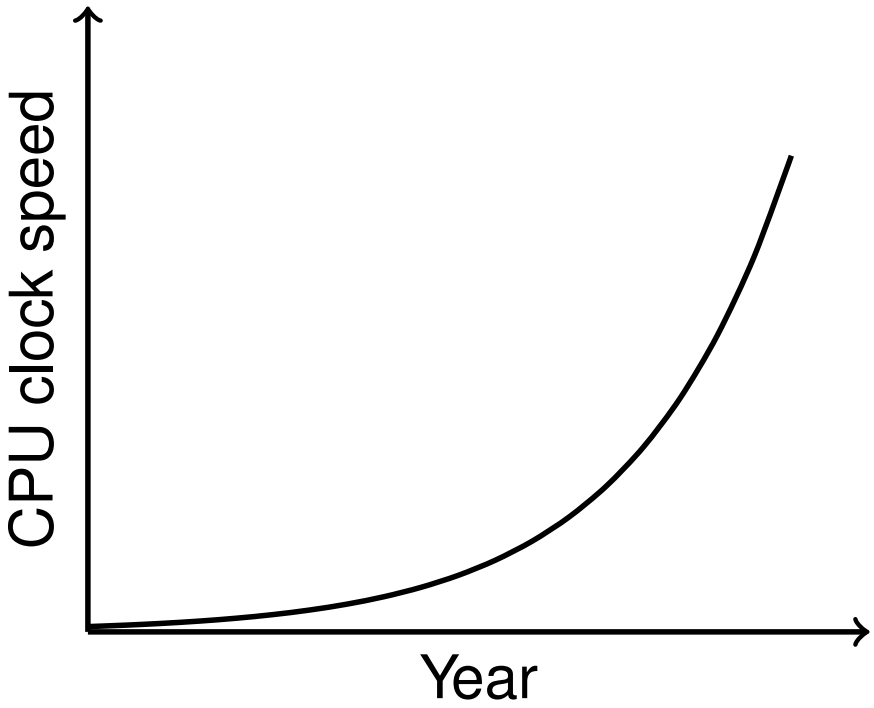
[2]Purdue University, USA

EuroSys 2007

Goal of the Talk

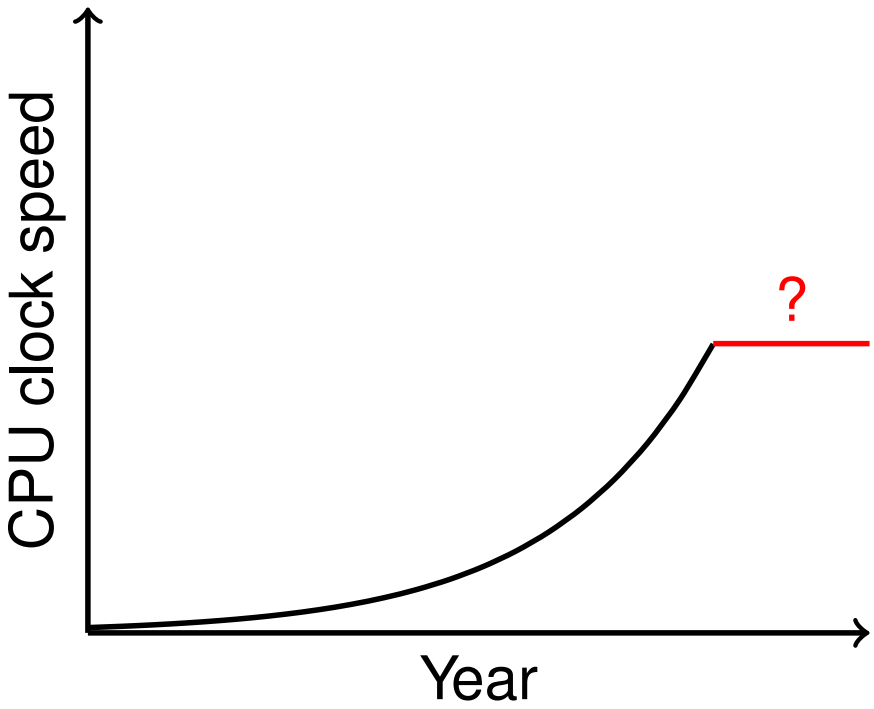
1. Recall the idea of software transactional memory
2. Present STMBench7: a benchmark for STM implementations

New Trends in Hardware



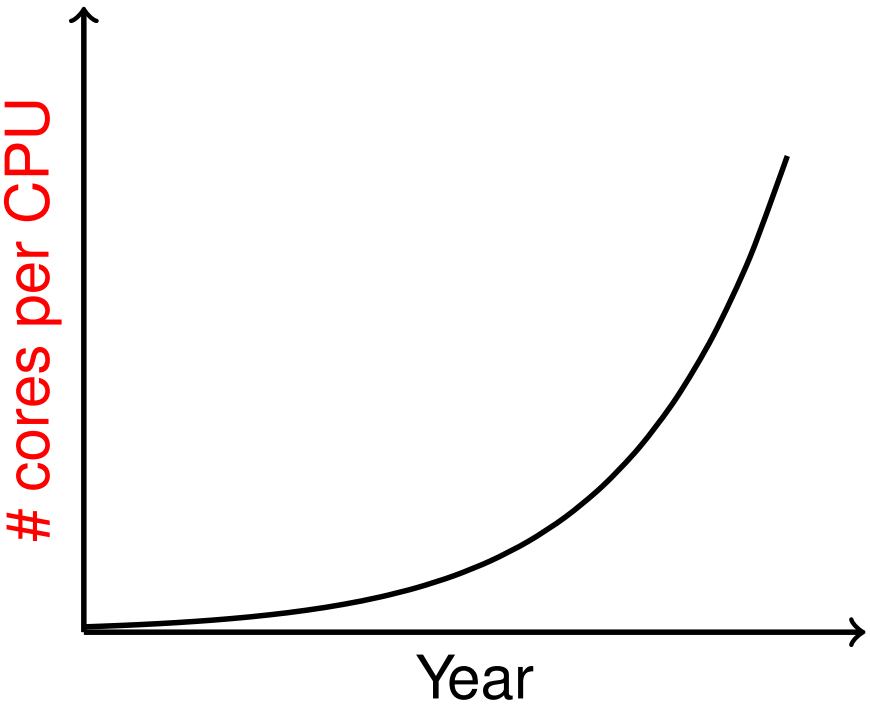
Common (mis)interpretation of Moore's law...

New Trends in Hardware



. . . now challenged

New Trends in Hardware



Solution: going multi-core

$\Downarrow$

New challenges for the masses of programmers:

1. Exploit parallelism
2. Manage concurrency

Current Approaches to Multi-threading

- Explicit locking is hard
  - Deadlock, priority inversion
  - Fault-tolerance issues

```
synchronized(this) {
  this.x.credit(5);
  this.y.debit(5);
}
```

- Wait-free/obstruction-free computing: not for mortals (each algorithm a PODC/DISC paper...)

Software Transactional Memory (STM)

- Multi-threading made easier: thread synchronization via in-memory transactions
- Does not share the inherent problems of locking

```
atomic {
  acc1.x.credit(5);
  acc2.y.debit(5);
}
```

$\Rightarrow$ commit or abort

STM implementations are coming!

TL2 RSTM Haskell STM

SI-STM OSTM

LSA-STM

DSTM

DSTM2 ASTM SXM ...

Which one is best?

Open Problems

1. Which implementation/algorithm is best?
2. What are the trade-offs?
3. Are the overheads acceptable?

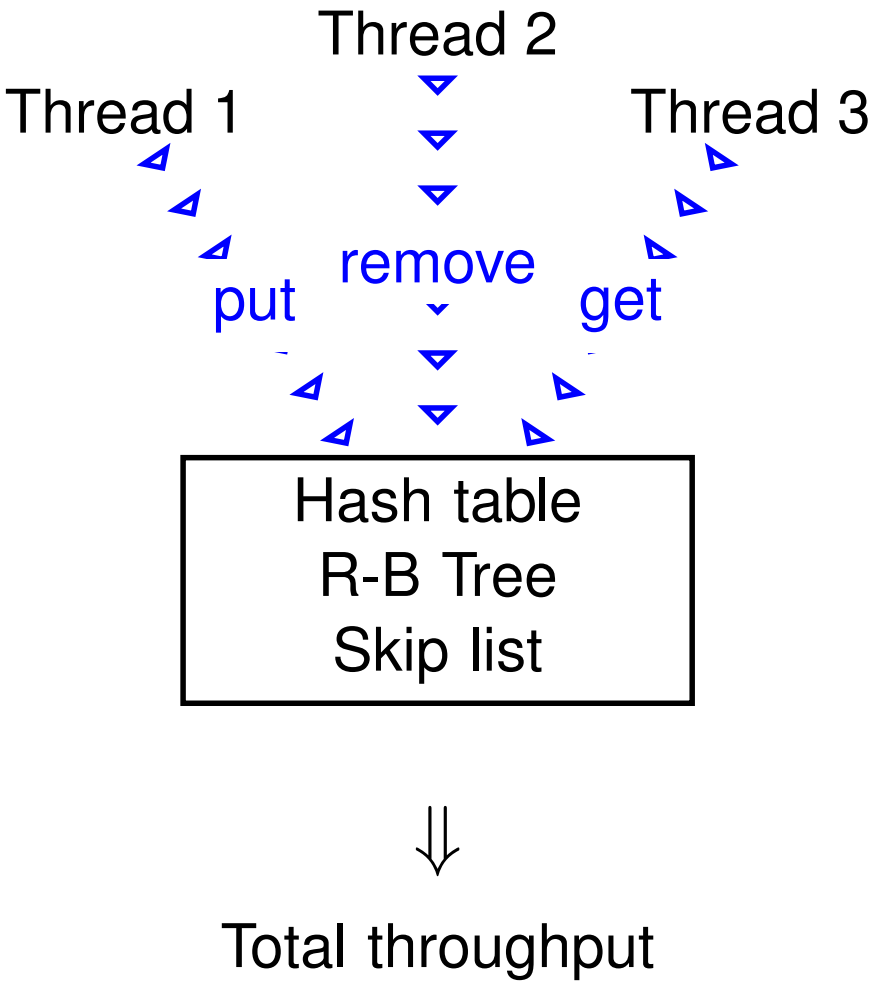
$\Rightarrow$ Need evaluation tools → theoretical

→ experimental $\Rightarrow$ STMBench7

Evaluation So Far

- Design features
- Simple data structures
- “Toy” applications

⇒ Need realistic benchmark



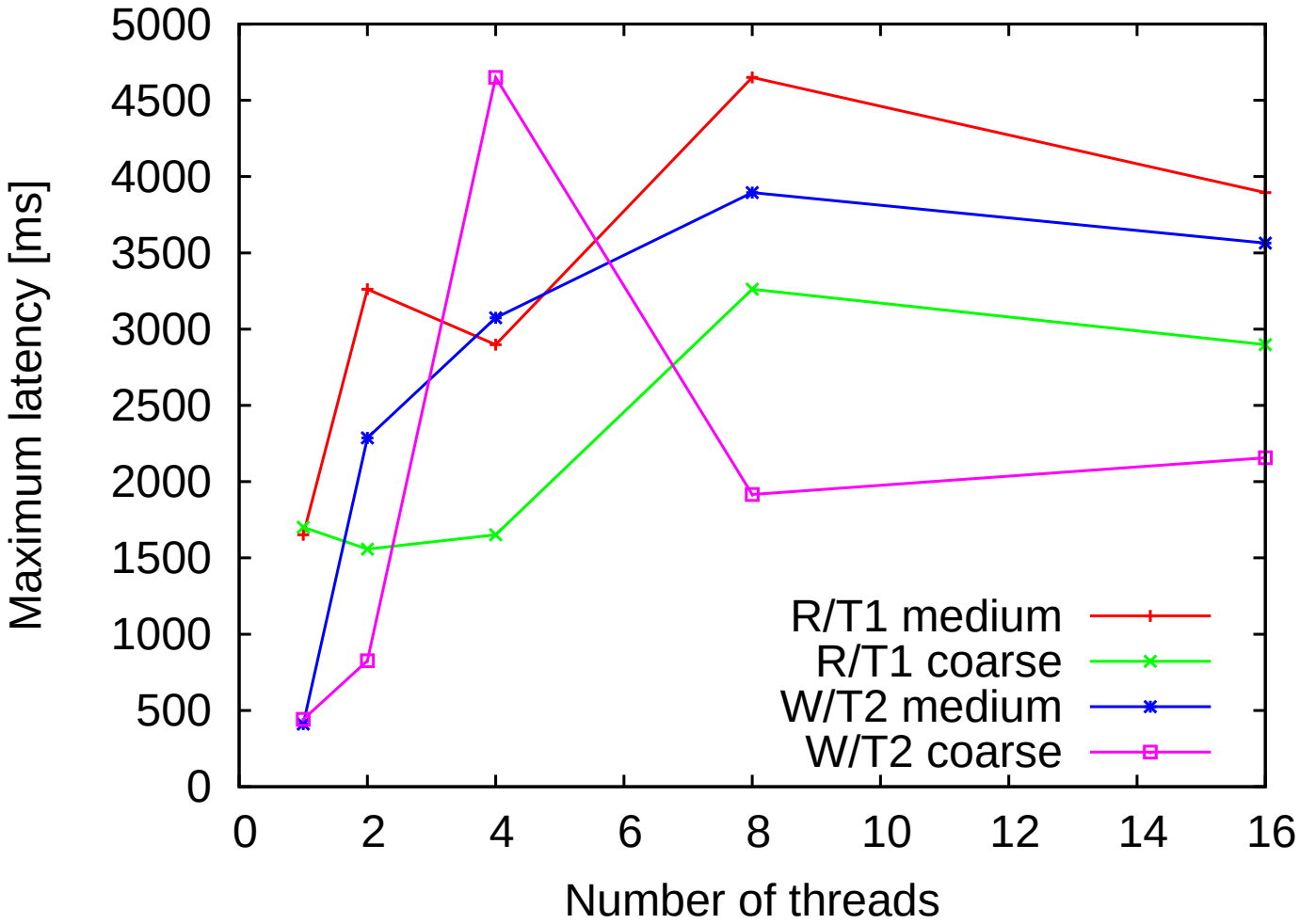
STM Benchmark Recipe

1. Realistic workload
2. Multi-threading
3. Non-trivial concurrency
4. Baseline for comparison

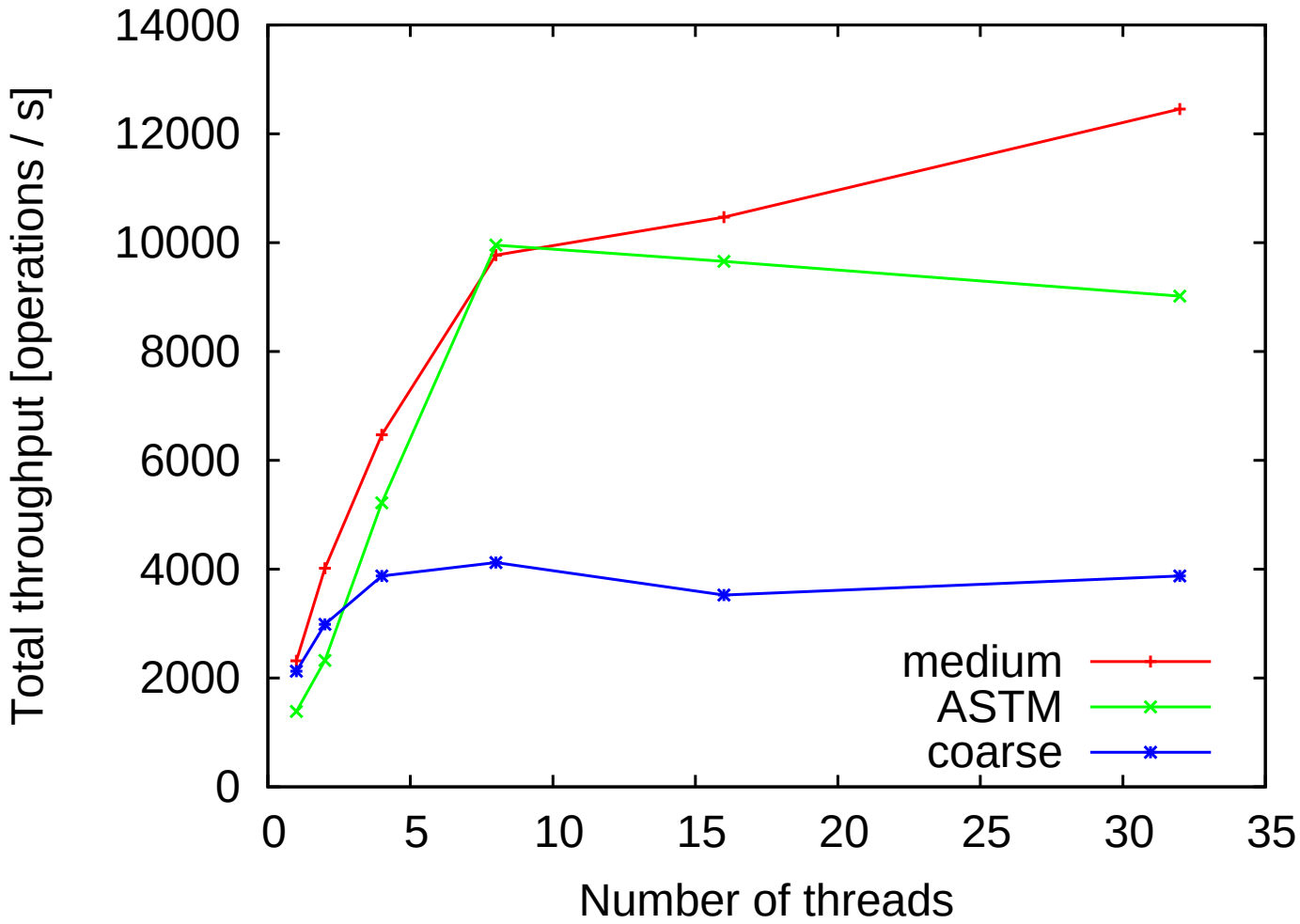
\+ std. benchmark requirements

STMBench7 is a first step towards such a benchmark

Example Output of STMBench7



Example Output of STMBench7



Realistic Workload

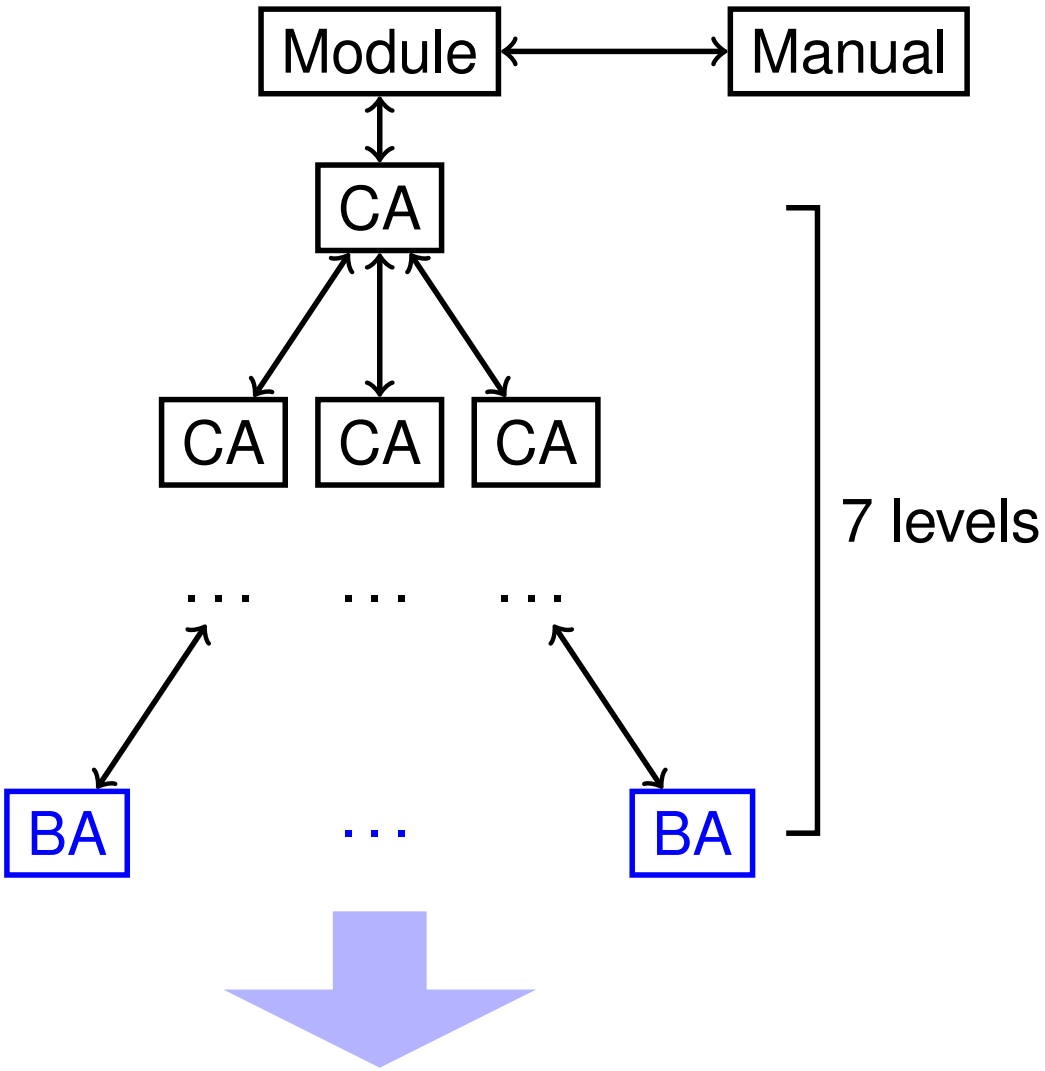
1. Realistic workload
2. Multi-threading
3. Non-trivial concurrency
4. Baseline for comparison

STMBench7 is based on the OO7 benchmark:

- Well-known in OO database world
- Aims at being realistic (CAD/CAM/CASE applications)
- Already used for transactional monitors

Way not enough!

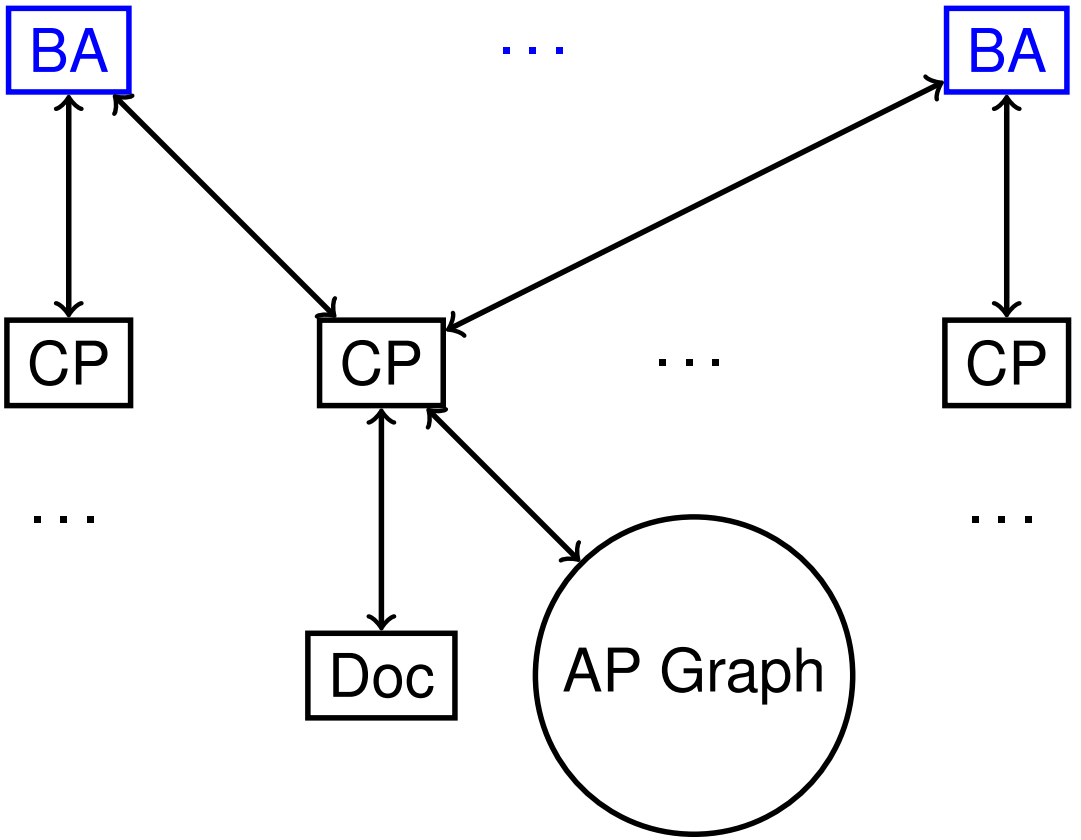
OO7 Data Structure (Main Tree)



Complex Assemblies (CA)

Base Assemblies (BA)

OO7 Data Structure (Design Library)



Base Assemblies (BA)

Design Library
(Composite Parts CP)

Atomic Parts (AP)

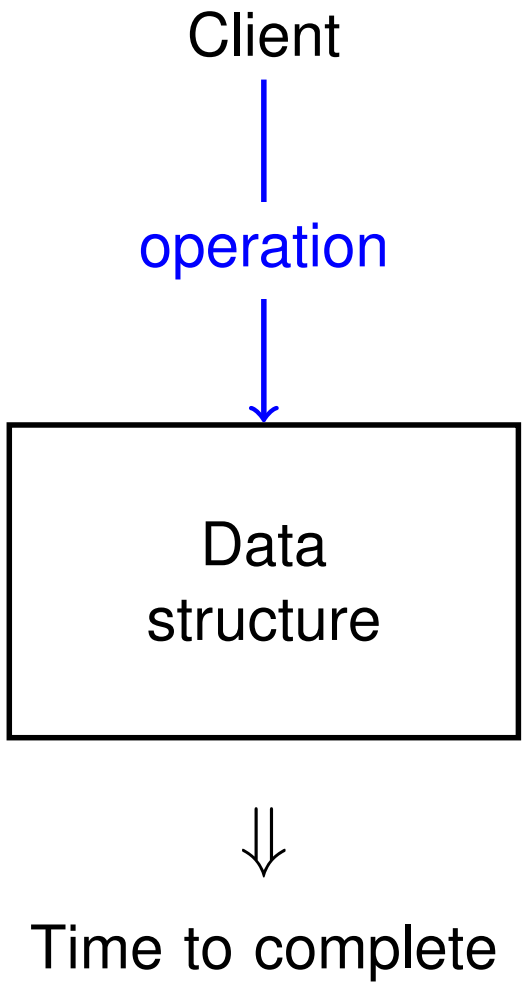
OO7 / STMBench7 Data Structure Summary

- Large tree with a graph in each leaf
- 6 indexes
- Can be traversed in any direction

OO7 Limitations

OO7:

- Single client
- TTC of isolated long operations measured
- Mostly static structure

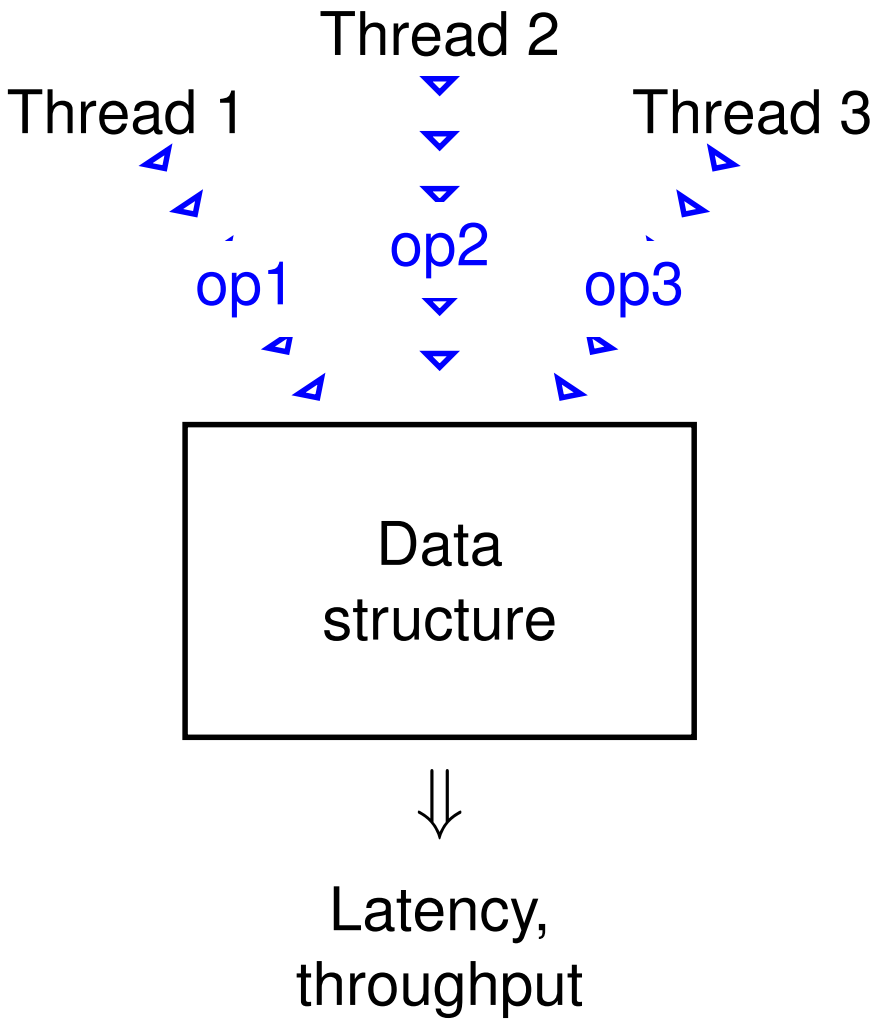


From OO7 to STMBench7

STMBench7:

- Multiple threads
- Various-length operations
- Dynamic structure

OO7-related code: $\sim$ 50%



Operation Types in STMBench7

- Traversals: access most objects
- Short traversals: access object on some random path
- Short operations: search and/or simple updates
- Structure modification operations

Operation Types in STMBench7

- Traversals: access most objects
- Short traversals: access object on some random path
- Short operations: search and/or simple updates
- Structure modification operations

1. Realistic workload
2. Multi-threading
3. Non-trivial concurrency
4. Baseline for comparison

Built-in Locking Techniques

1. Realistic workload
2. Multi-threading
3. Non-trivial concurrency
4. Baseline for comparison

Built-in Locking Techniques

1. Realistic workload
2. Multi-threading
3. Non-trivial concurrency
4. Baseline for comparison

- Coarse-grained locking: single RW lock
- Medium-grained locking:
  - one RW lock per level,
  - global RW lock for structure modifications.

Porting Issues

Languages:

- Current implementation in Java 5 ($\sim$ 5000 lines)
- C# port being developed (group of M. Herlihy)
- C++ version coming soon (with M. Scott)

STMs:

- Experiments done with ASTM
- Easy to plug other STMs
  (although no single STM interface)

Summary

- STMBench7 is a first attempt at realistic benchmarking STMs
- Data structure suggestive of CAD, CAM, CASE and similar applications, but also some on-line services
- Code already available (and ready for use)
- Open-source (BSD license)
- First experiments show that STMBench7 is a “stress test” for STM (see the paper)

Open Problems (Recalled)

1. Which implementation/algorithm is best?
2. What are the trade-offs?
3. Are the overheads acceptable?

$\Rightarrow$ Need evaluation tools → theoretical (stay tuned...)

→ experimental $\Rightarrow$ STMBench7

Open Problems (Recalled)

1. Which implementation/algorithm is best?
2. What are the trade-offs?
3. Are the overheads acceptable?

$\Rightarrow$ Need evaluation tools → theoretical (stay tuned...)

→ experimental $\Rightarrow$ STMBench7

Future directions:

- Add fine-grained locking
- Data validation

Questions?

lpd.epfl.ch/kapalka/stmbench7.php