Minor Thesis

Development and Analysis of Barrier Protocols

Ronny Brendel (http://automaton2000.com)
Responsible Professor: Prof. Dr. Christel Baier
Supervisor: Dr. Sascha Klüppelholz

Dresden, 2014-01-08

Content

- Introduction

    Basics, Motivation

- Protocols

    Central Counter, B1 Barrier

- Modelling

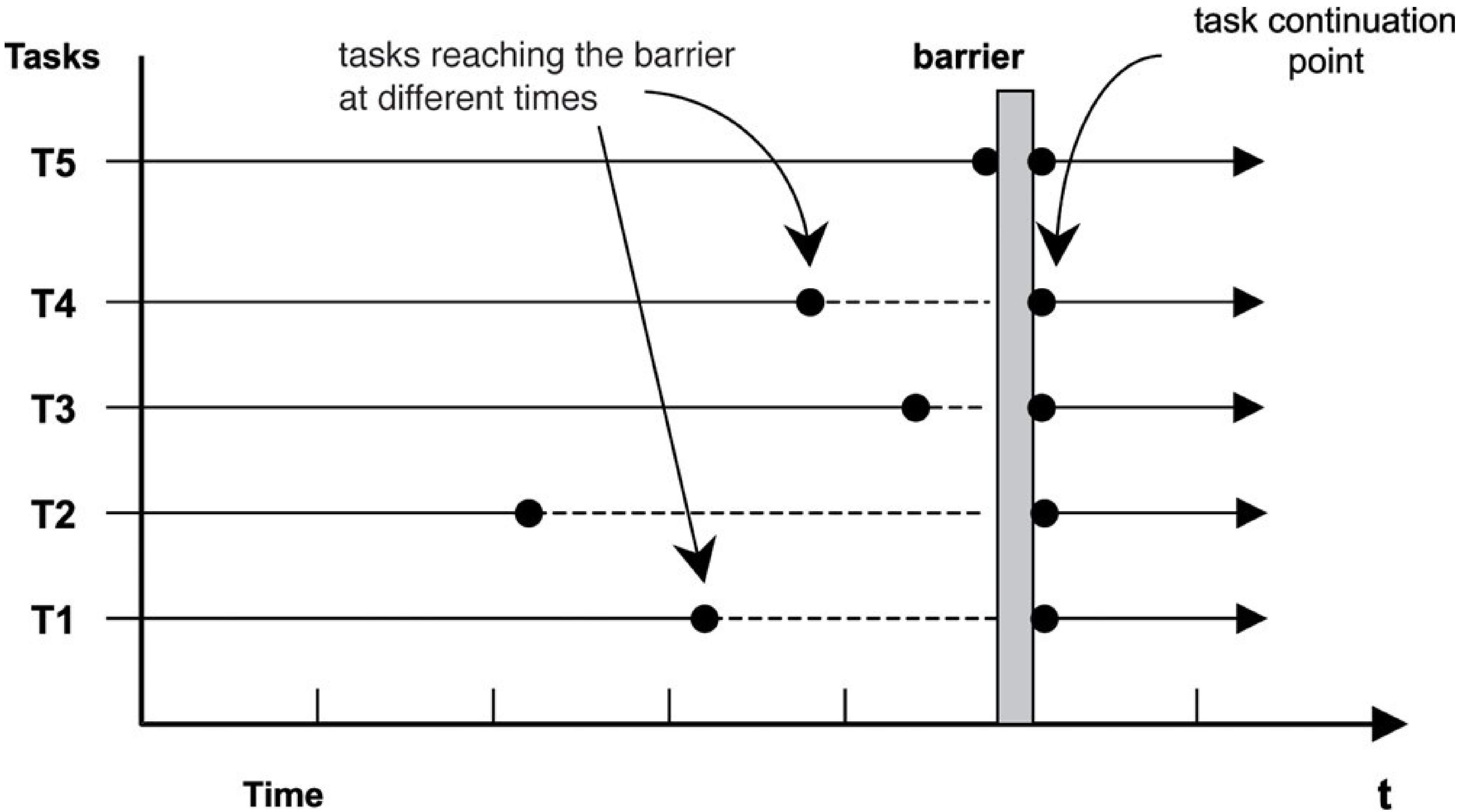
    Shared Memory, Protocols

- Analysis

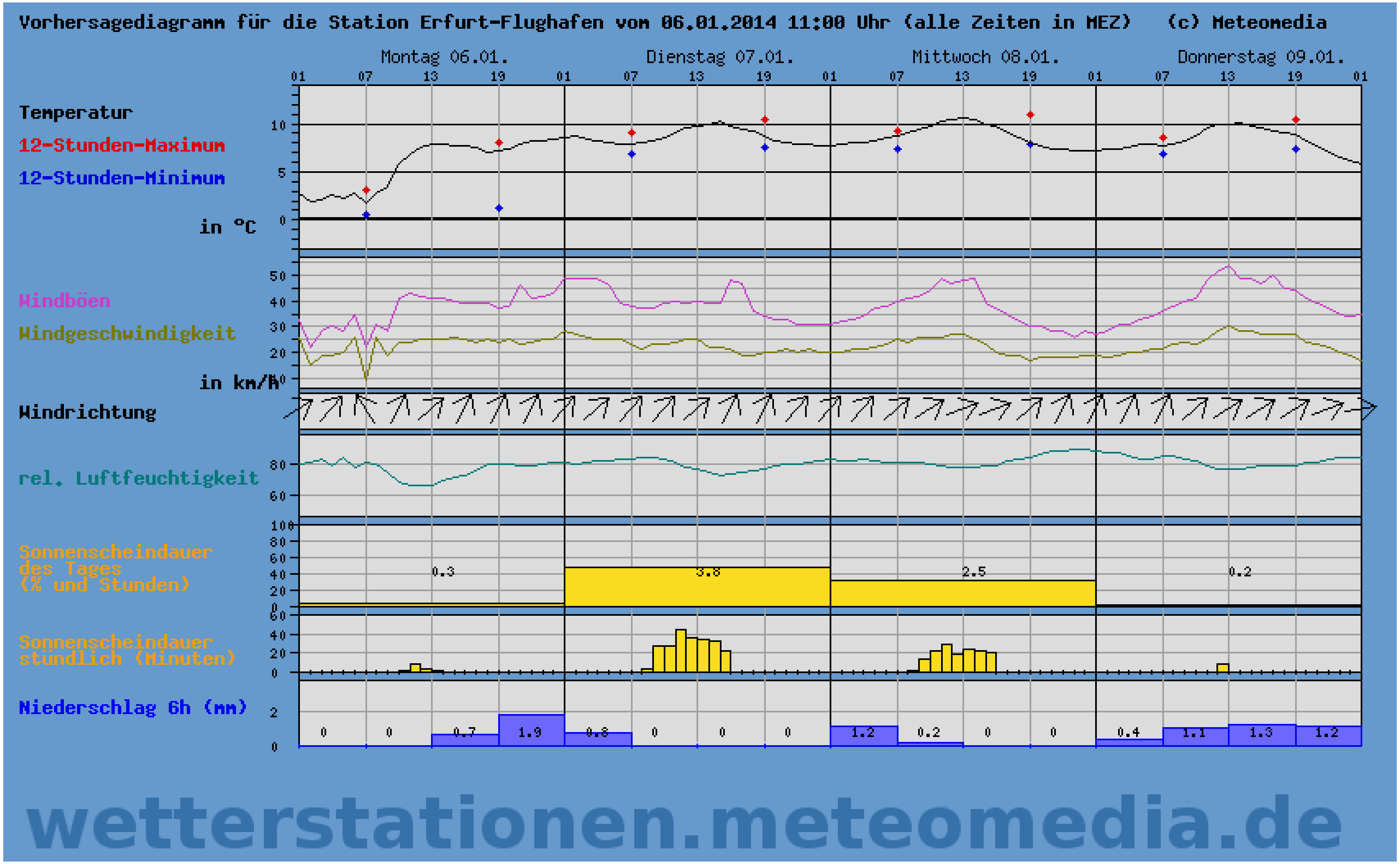
    Functional, Quantitative

- Conclusion, Future Work, Sources

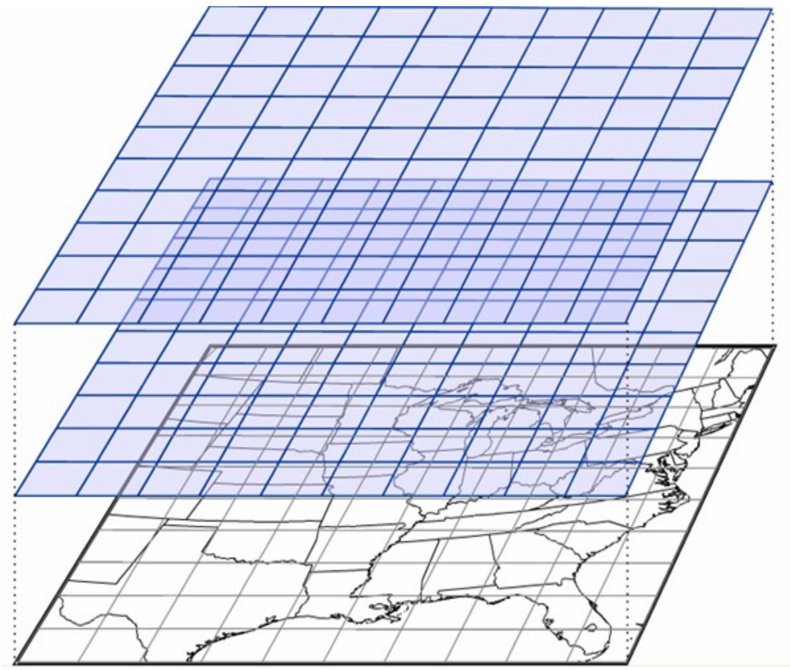
Basics < Introduction

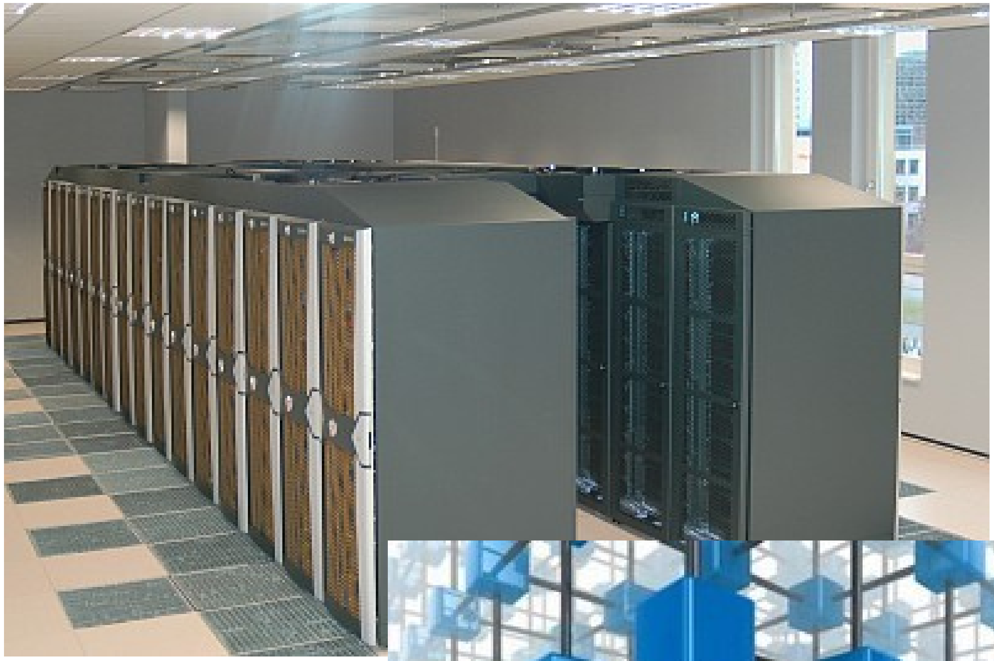


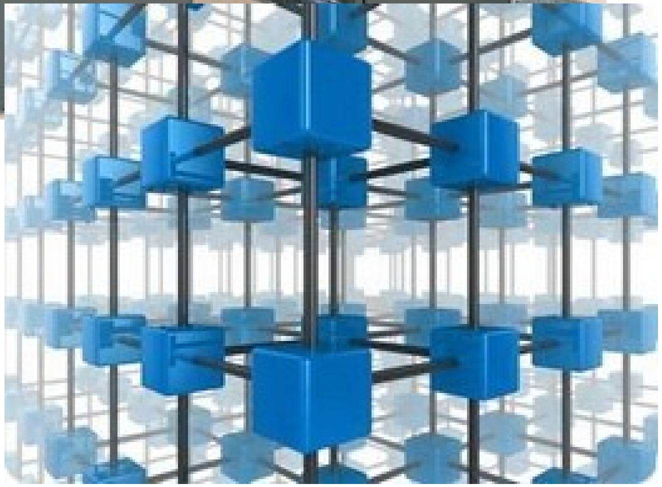
Basics < Introduction



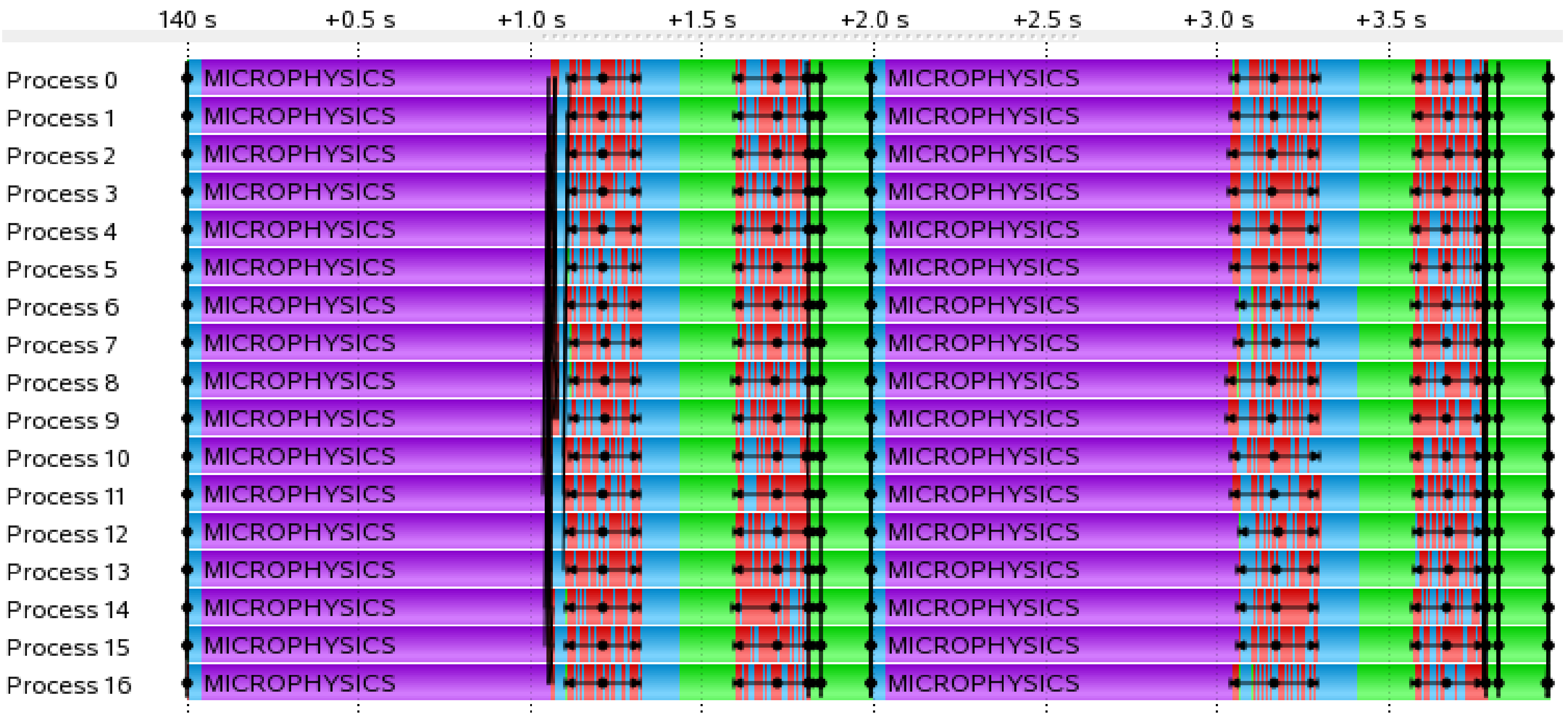
Basics < Introduction





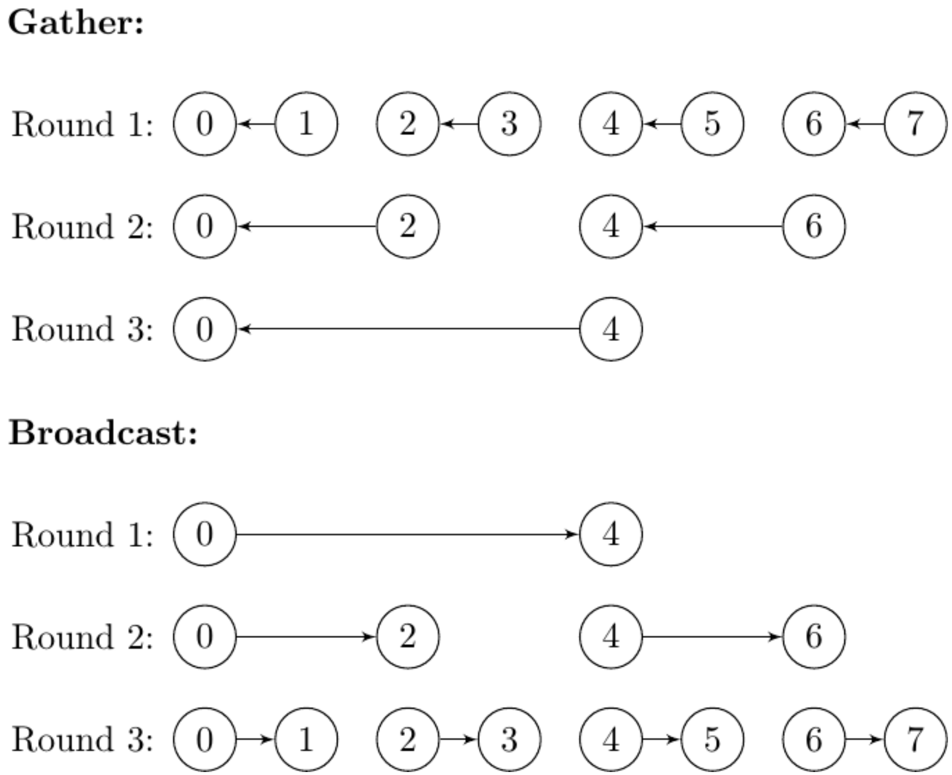


Basics < Introduction



Basics < Introduction

- Usual Implementations include
  - Central Counter Barrier (atomic increment)
  - Hierarchical approaches



see sources [3]

Motivation < Introduction

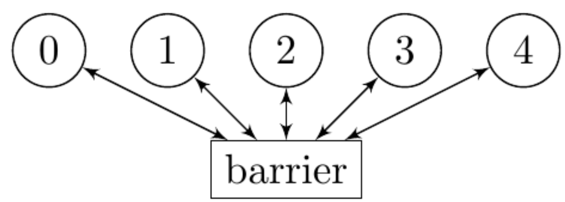
- Today's Barrier Protocols have been invented long ago
- Probabilistic Write/Copy-Select (pW/CS)
  - Concurrent protocols are unnecessarily strict
  - Relieving strictness can improve performance
  - Complexity of modern computers makes the timing of concurrent interaction effectively random. Employ the tools of probability theory for designing protocols

see sources [1,2]

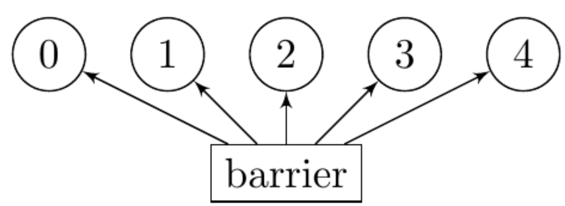
Central Counter Bar. < Protocols

```
shared variables: integer barrier := threadCount
----------------------------------------------------

atomic{barrier := barrier − 1}

wait until barrier = 0
```

Atomic decrement:



Repeated reading:



B1 Barrier < Protocols

```
shared variables: boolean barrier[threadCount]
local variables:  integer i
initialisation:   barrier[*] := false
---------------------------------------------

barrier[threadIndex] := true

i := 0
while i < threadCount {
    if barrier[i] = false {
        i := -1
    }
    i = i + 1
}
```

Modelling

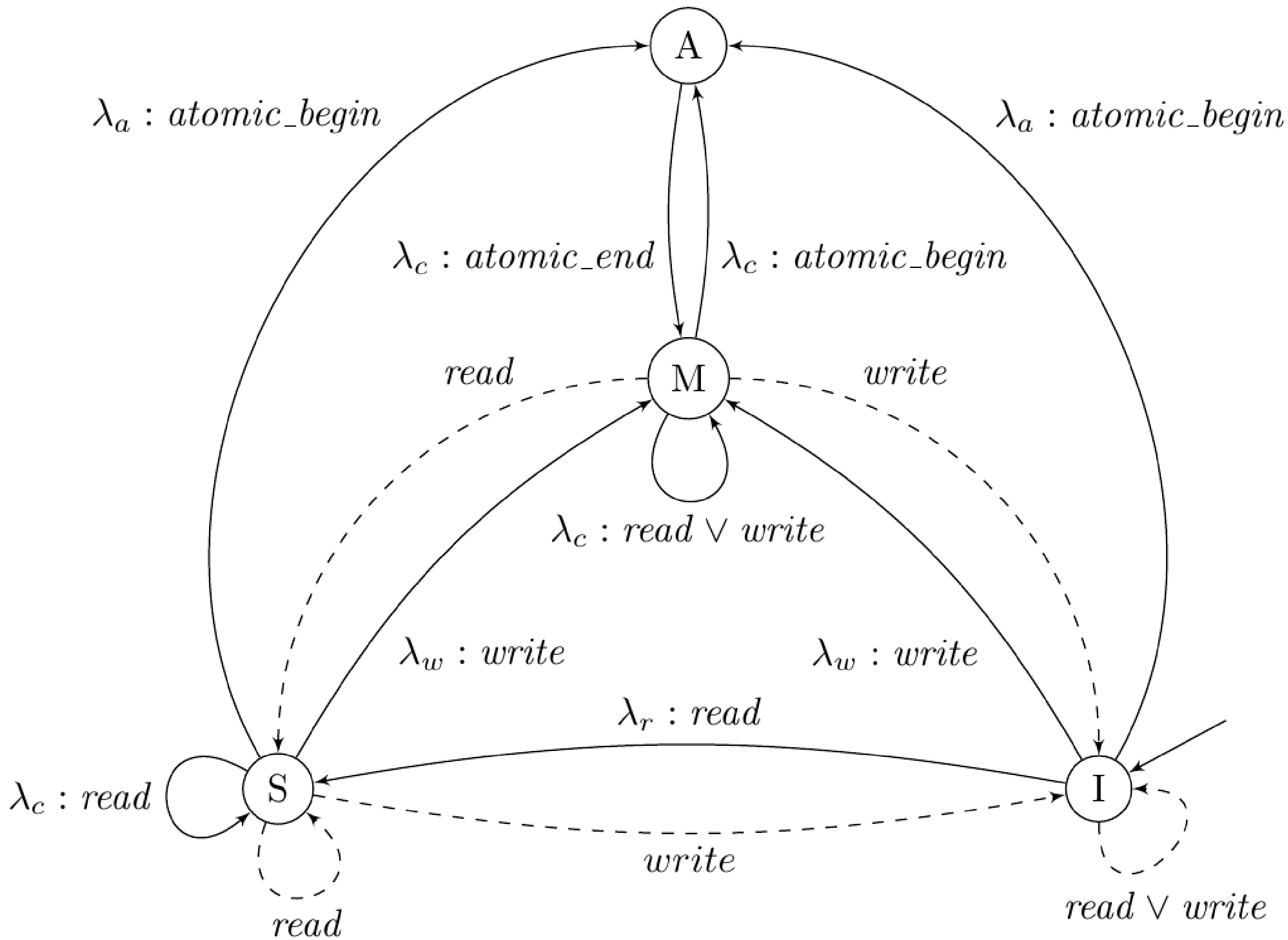
- Functional
  - Non-deterministic transition system + LTL
  - SPIN
  - detailed model to reveal all possible mistakes
- Quantitative
  - CTMC + CSL/CSRL
  - PRISM
  - reduced to just costly transitions, no reinitialisation

see sources [4,5]

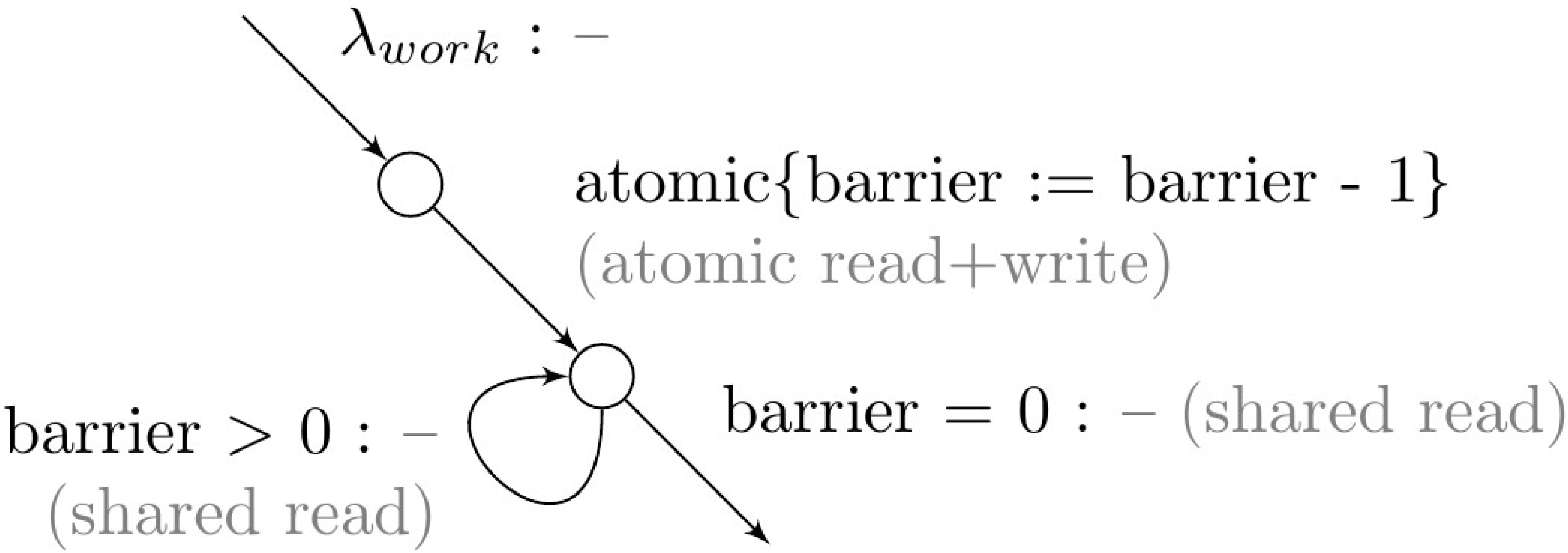
Shared Variable < Modelling

- Synchronisation is about exchanging information, i.e. sharing memory
- Very small information -> Timing dominated by memory access latency
- Memory access is cached -> We have to model caching
- We identify a shared variable with the cache line it resides on
- MSI protocol + atomic operations

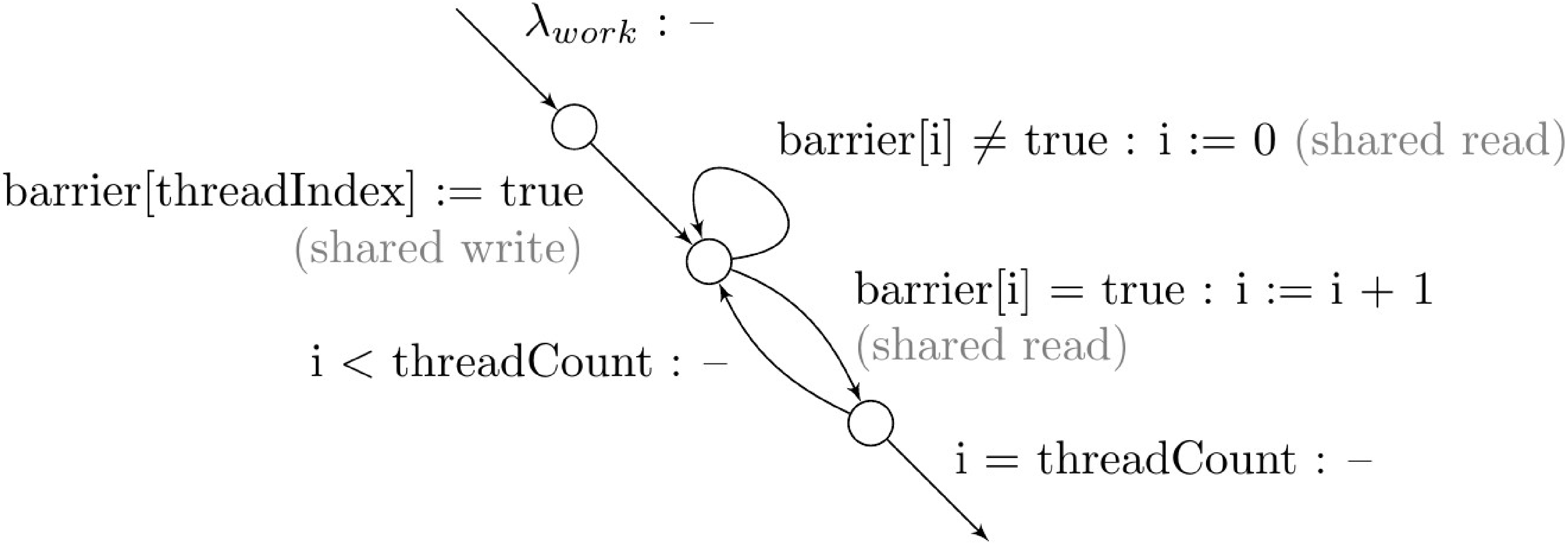
Shared Variable < Modelling



Central Counter Bar. < Modelling



B1 Barrier < Modelling



Functional < Analysis

Functional < Analysis

Quantitative < Analysis

Conclusion

- Introduced innovative barrier protocols
  - + No atomic operations or locks required
  - + Competitive performance
  - – Bandwidth/Energy hungry
- Principles of pW/CS apt to improve synchronisation performance
- Quantitative model checking enables exhaustive, fine-grained analysis beyond the capability of tests/benchmarks

Future Work

- Analyse protocols using measurement
- Invent more protocols
  - variations of existing
  - remote write-based
- Extend model checking
  - more processes/threads
  - more detail
    - cache protocols, cache hierarchies
    - limited bandwidth and other influences

Sources

- [1] Probabilistic write copy select, Paper,

  In 13th Real-Time Linux Workshop, pages 195–206, Oct. 2011

- [2] A probabilistic quantitative analysis of probabilistic-write/copy-select, Paper,

  In NASA Formal Methods, pages 307–321. Springer, 2013.

- [3] Evaluation of publicly available Barrier-Algorithms and Improvement of the Barrier-Operation for large-scale Cluster-Systems with special Attention on InfiniBand Networks

  http://htor.inf.ethz.ch/publications/index.php?pub=12

- [4] PRISM, Website, 13-03-019

  http://www.prismmodelchecker.org

- [5] SPIN, Website, 13-01-08

  http://spinroot.com

Thank you!

Slides and report are available at
http://automaton2000.com/barrier-slides.pdf
http://automaton2000.com/barrier-minor-thesis.pdf