Modeling Multiprocessor Cache Protocol Impact on MPI Performance

Ghassan Chehaibar*, Meriem Zidouni*†, Radu Mateescu†

*Bull SAS - †INRIA

AINA'09, QuEST Workshop, Bradford, May 26-29, 2009

Motivation

- **Bull HPC servers**
- **MPI library.**
- **Cache-coherent distributed shared memory nodes.**

Motivation

- **Bull HPC servers**
- **MPI library.**
- **Cache-coherent distributed shared memory nodes.**

Ping-Pong benchmark

?

Measured latency not conform to miss count expectations

Motivation

- Bull HPC servers
- MPI library.
- Cache-coherent distributed shared memory nodes.

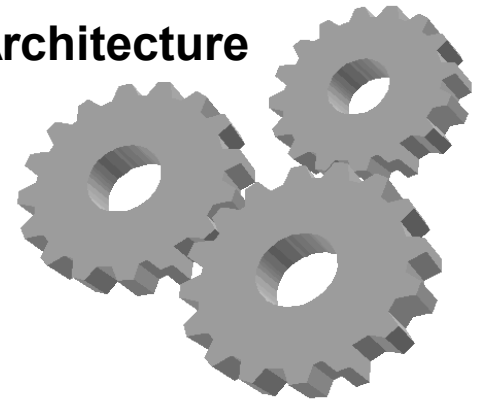
Ping-Pong benchmark

?

Measured latency not conform to miss count expectations

Software

Architecture



Cache protocol

Need a method to correctly evaluate latency and miss count per variable

Interaction between benchmark software, cache protocol, and architecture topology

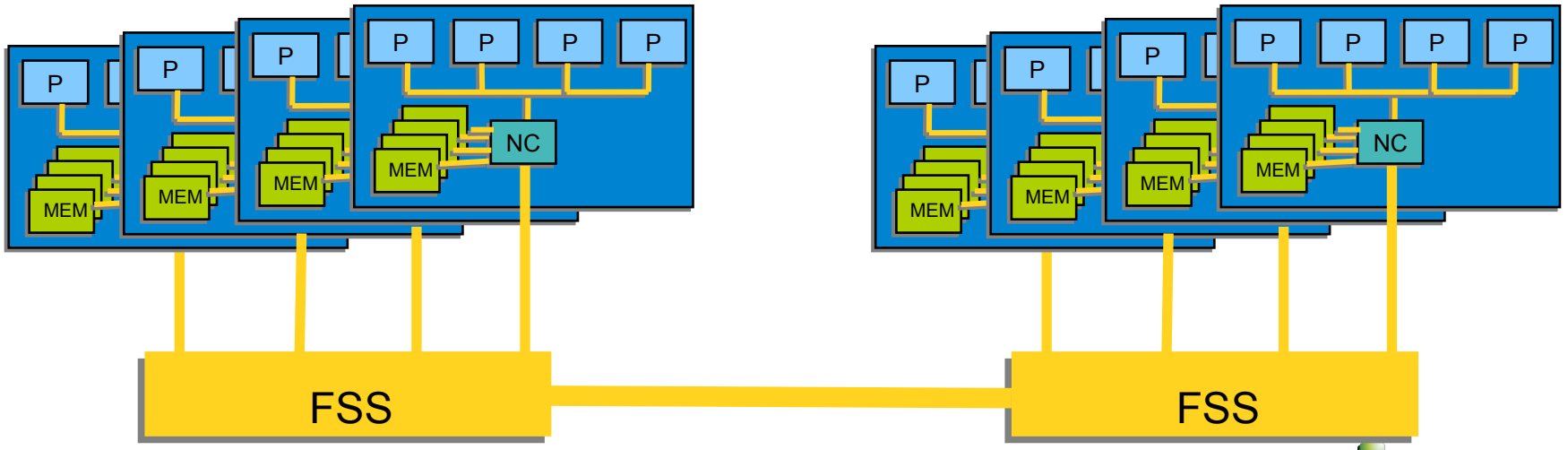
Related work

- Measurement:
  - Post hardware development phase.
  - Lack of analysis elements without complex instrumentation (like miss count per variable).
- Simulation environment:
  - Complex model construction with libraries and C-code snippets.
  - All the more, we consider parallel processes.
  - No way to verify correctness of modeling.
- Not aware of an already published work that:
  - Allows verifiable modeling interactions of complex aspects like cache coherence protocol, architecture topology and software algorithm.
  - Provides not only overall performance figures but also analysis elements.

Using formal methods

- Formal modeling of the functional behavior with LOTOS.
- Formal verification of model correctness with CADP toolbox.
- Integration of performance aspects based on *Interactive Markov Chain* theory: smooth extension of LOTOS.
- Generation of a Continuous Time Markov Chain, which we are confident properly reflects functionality and performance behaviors.
- Use of numerical analysis algorithms to calculate relevant performance figures.

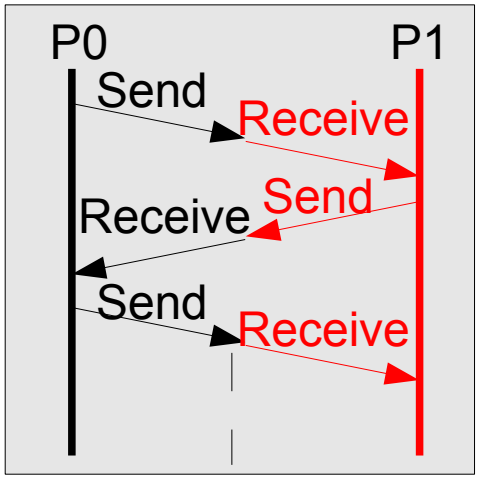
System at hand



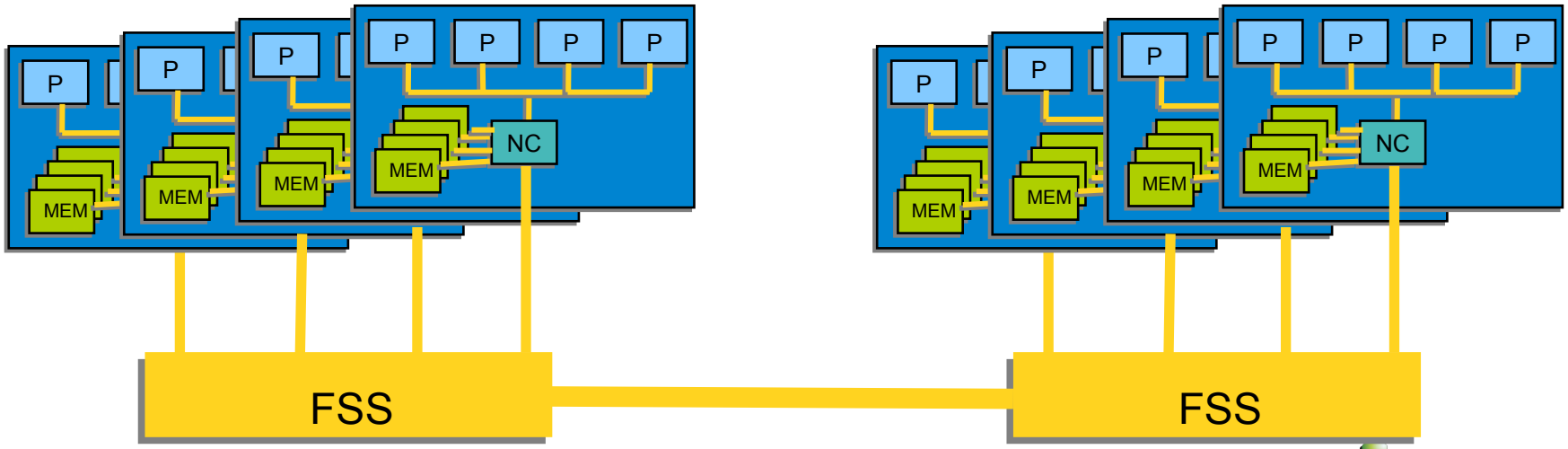
Architecture

System at hand

Ping-Pong benchmark **(latency of send;receive)**



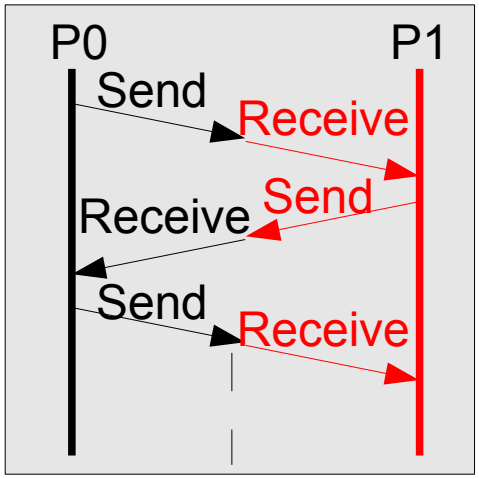
Mapping



Architecture

System at hand

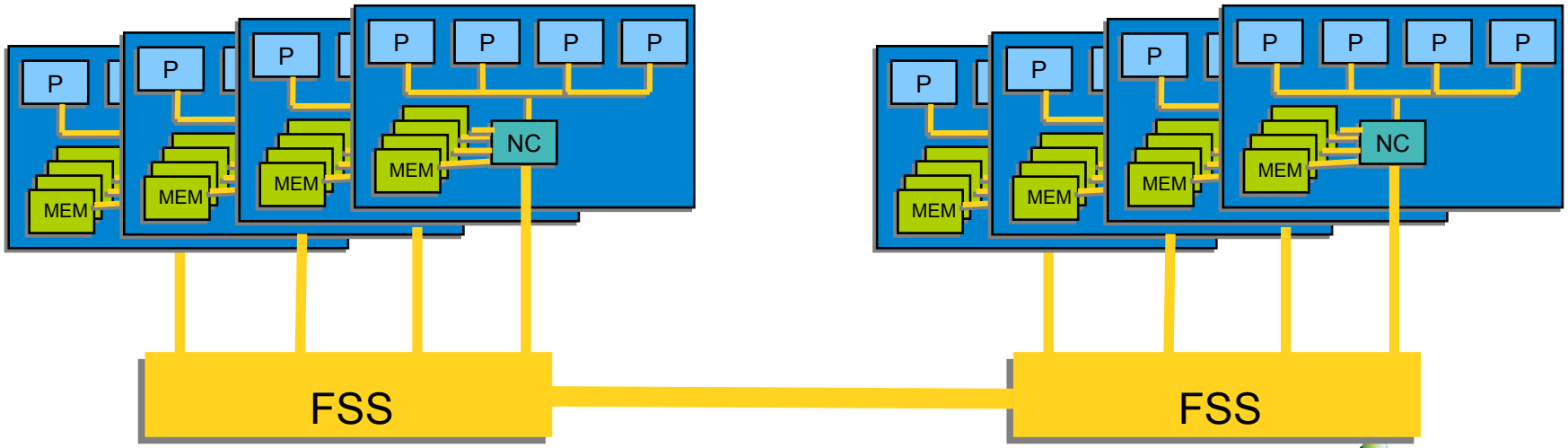
Ping-Pong benchmark (latency of send;receive)



Mapping

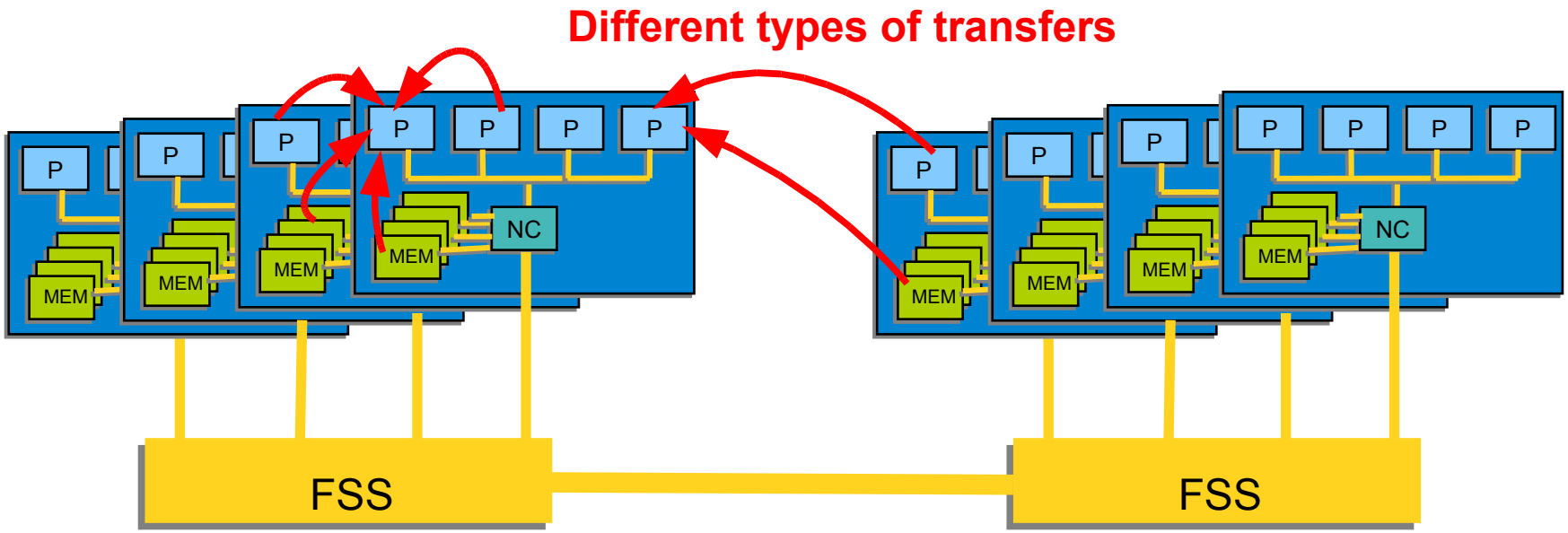
Cache coherence protocol

- Hit/Miss
- Cache state changes
- Data transfer types



Architecture

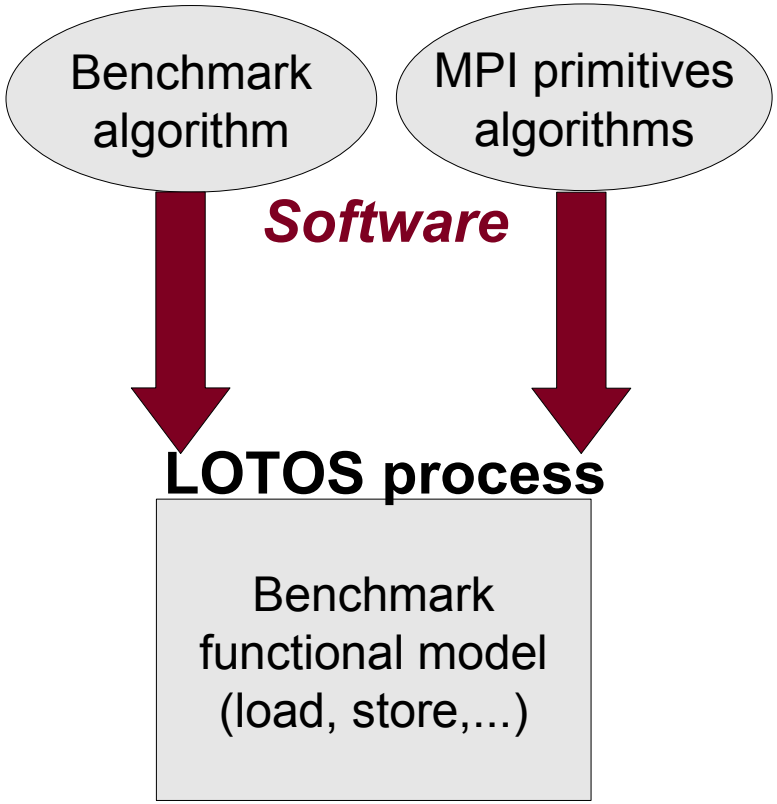
State-dependent latency



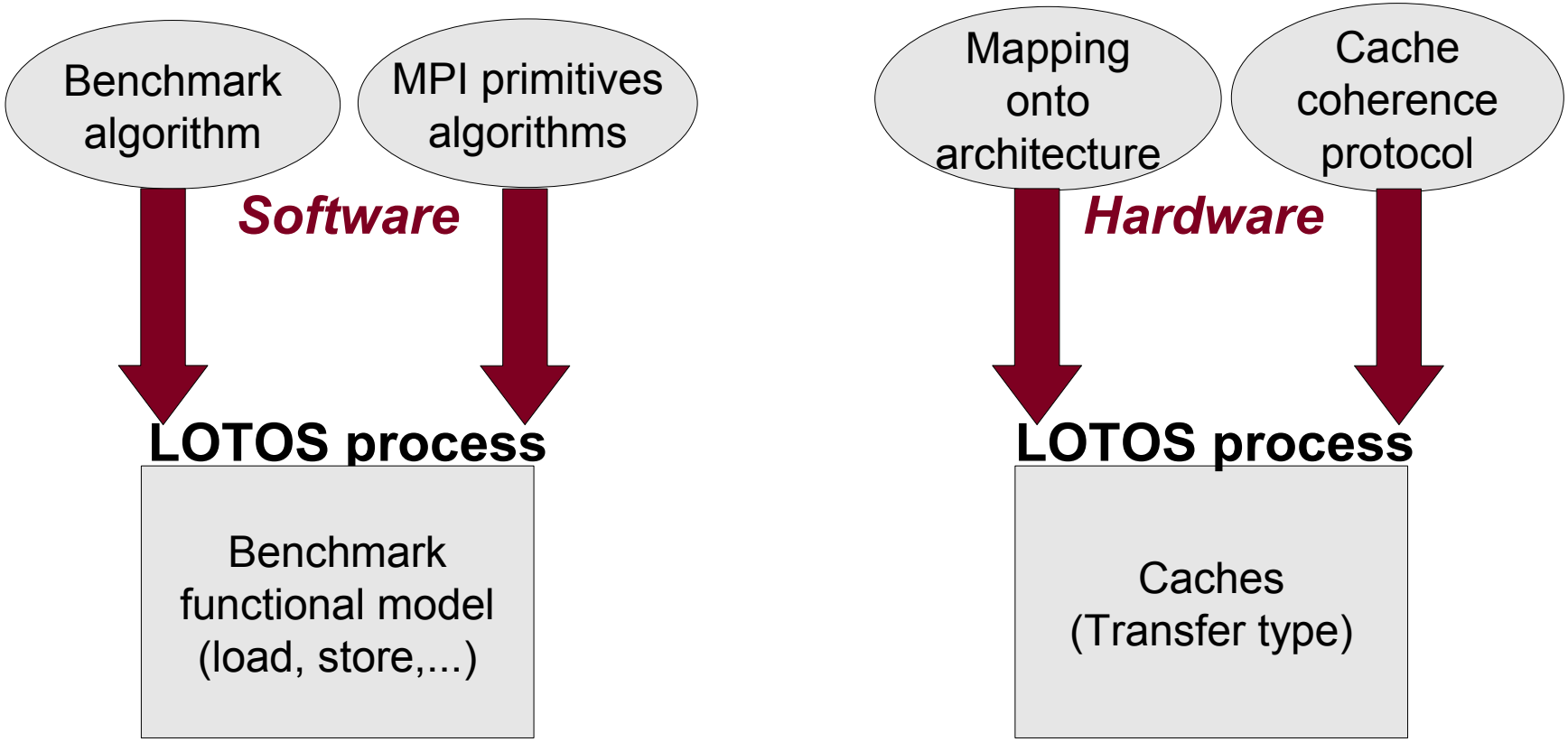
- Latency of an access: depends on dist(requester, data).
- Data location at a given time: depends on cache protocol and history execution of parallel processes.

Latency of an access depends on global state of caches in the system

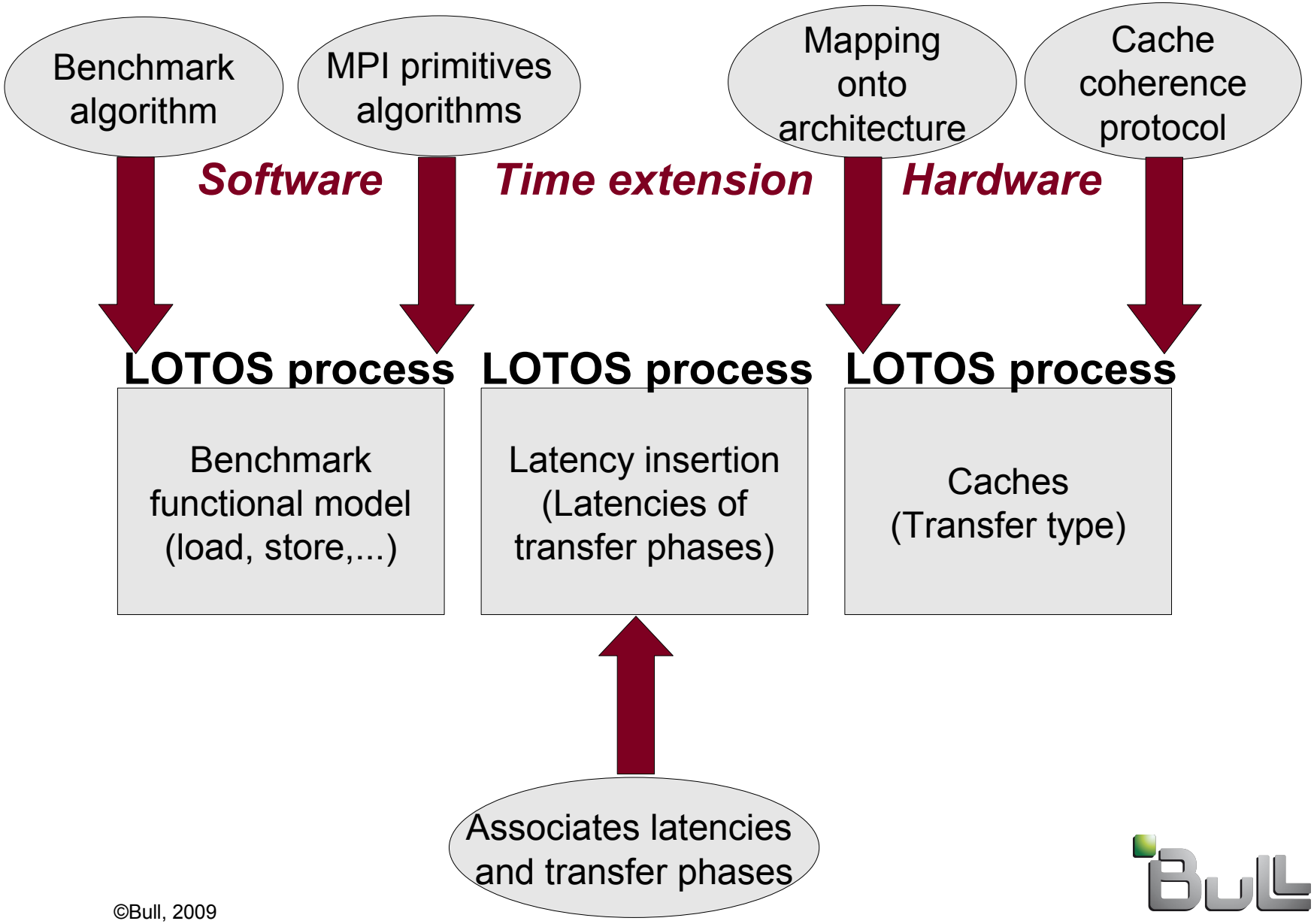
Separation of concerns



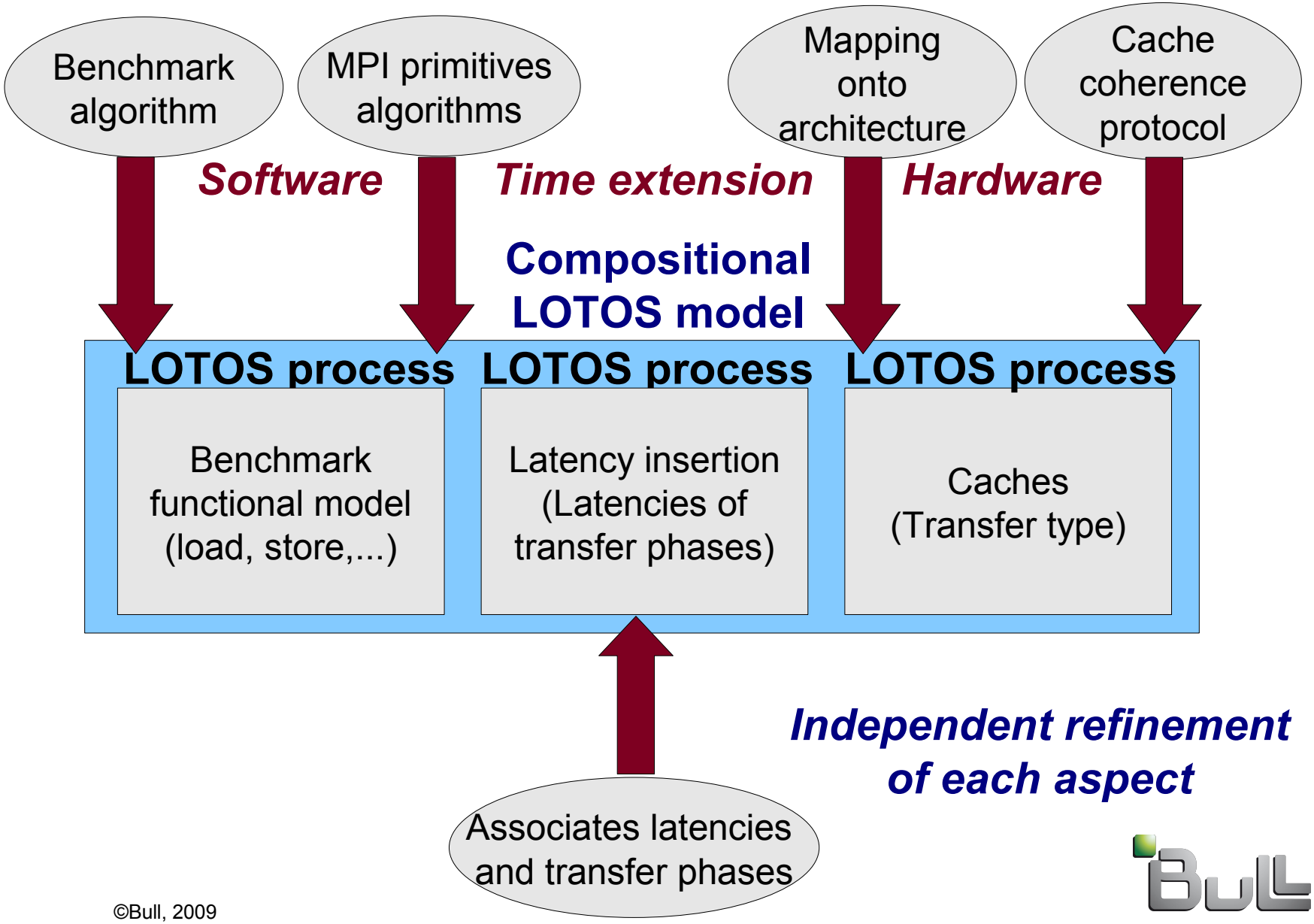
Separation of concerns



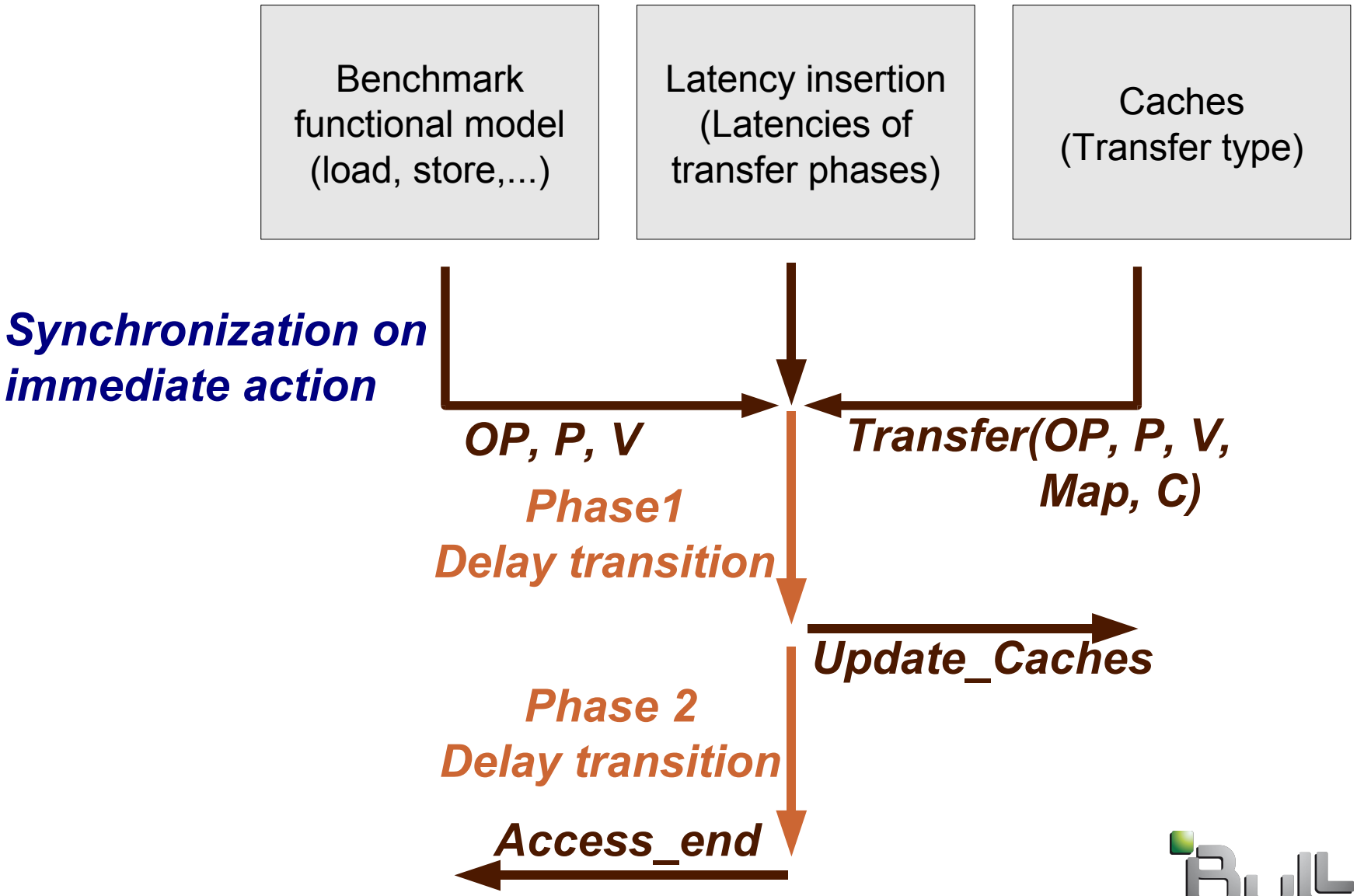
Separation of concerns



Separation of concerns



Modeling state-dependent latencies



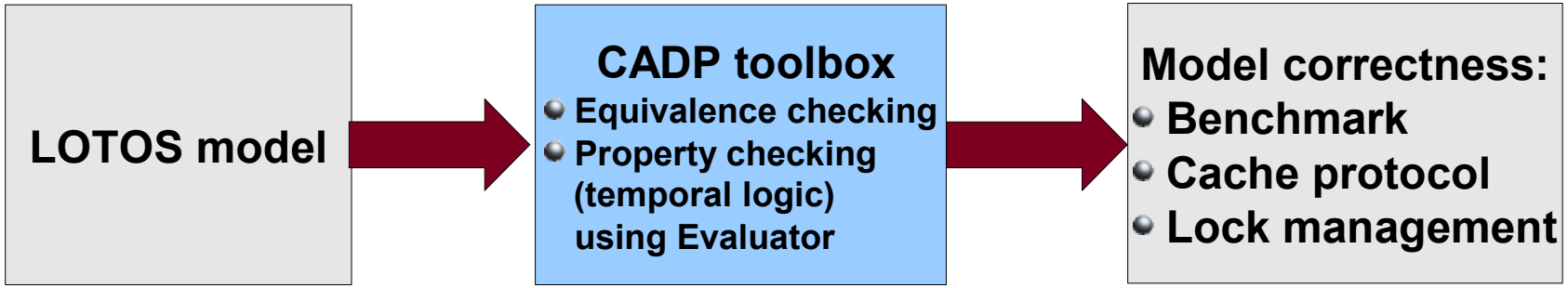
Formal verification / Performance analysis

LOTOS model

Same model for functional and performance behavior

Same tool and technique for formal verification and performance evaluation

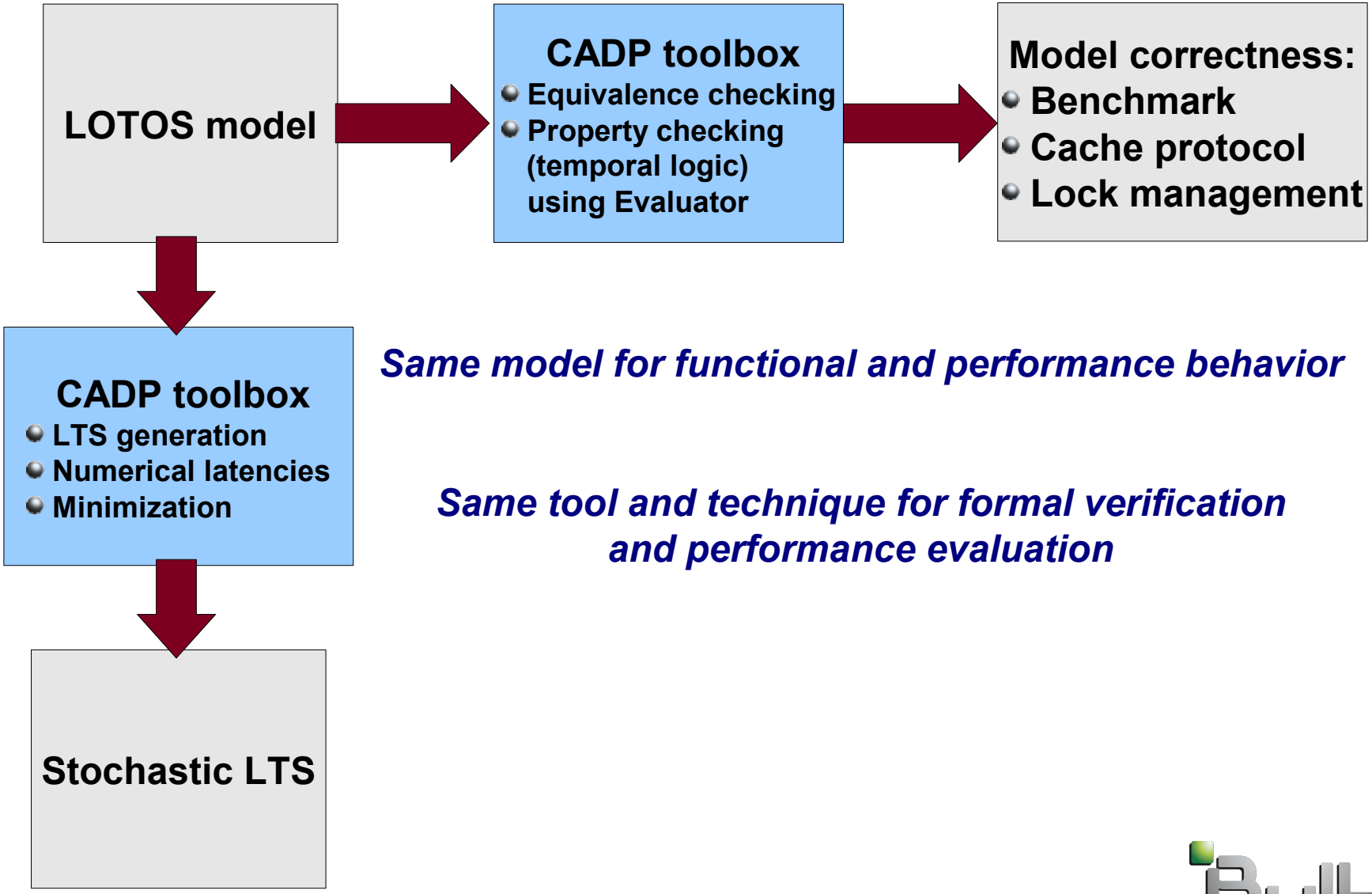
Formal verification / Performance analysis



Same model for functional and performance behavior

Same tool and technique for formal verification and performance evaluation

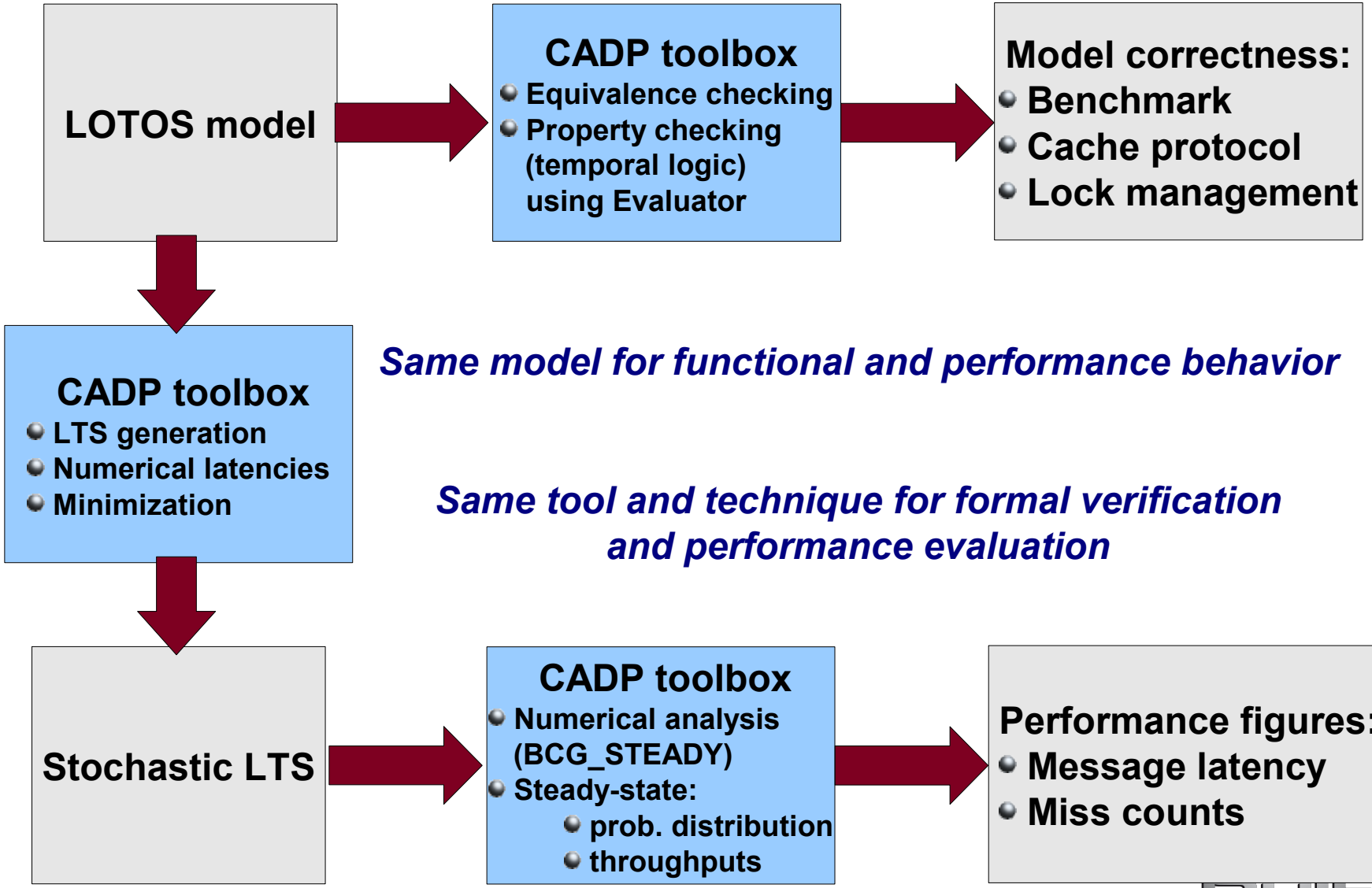
Formal verification / Performance analysis



Same model for functional and performance behavior

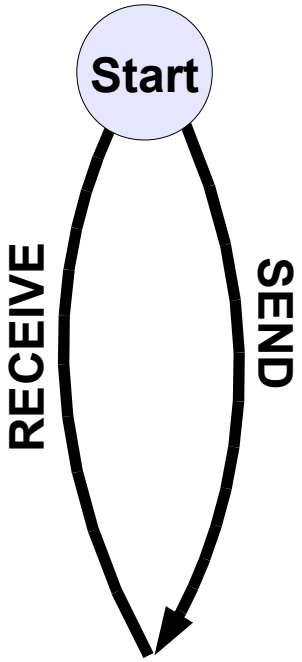
Same tool and technique for formal verification and performance evaluation

Formal verification / Performance analysis



Computing performance figures

Message exchange latency

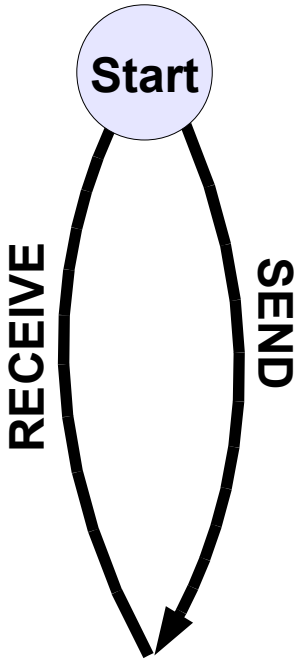


Average loop latency =

$$\frac{1}{throughput(Start)}$$

Computing performance figures

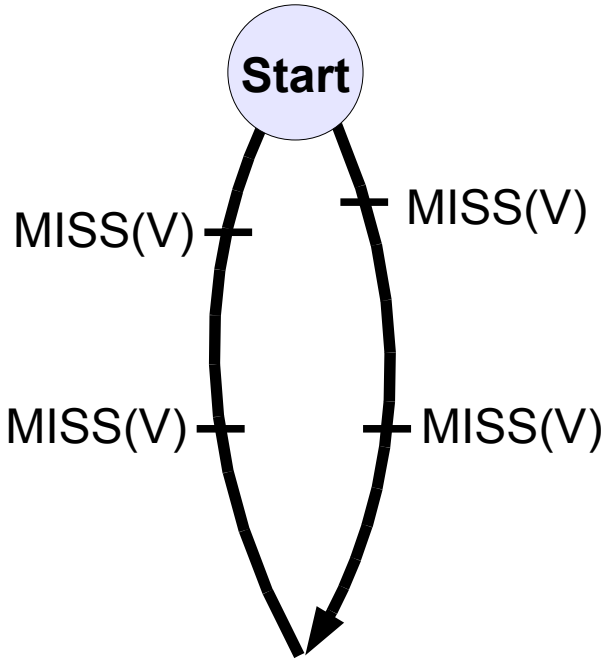
Message exchange latency



Average loop latency =

$$\frac{1}{throughput(Start)}$$

Miss count during an exchange



Average miss count =

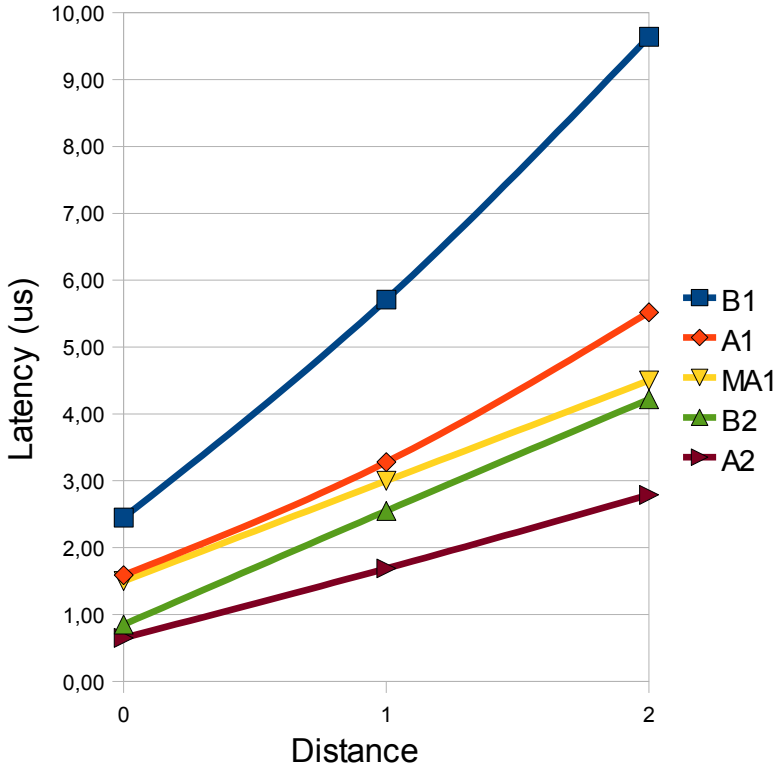
$$\frac{throughput(MISS(V))}{throughput(Start)}$$

Evaluation results in several configurations

- Mapping of processes onto the architecture → distance.
- Implementation of primitives (1, 2)
- Cache coherence protocol variant: A, B.

Evaluation results in several configurations

- Mapping of processes on the architecture → distance.
- Implementation of primitives (1, 2)
- Cache coherence protocol variant: A, B.

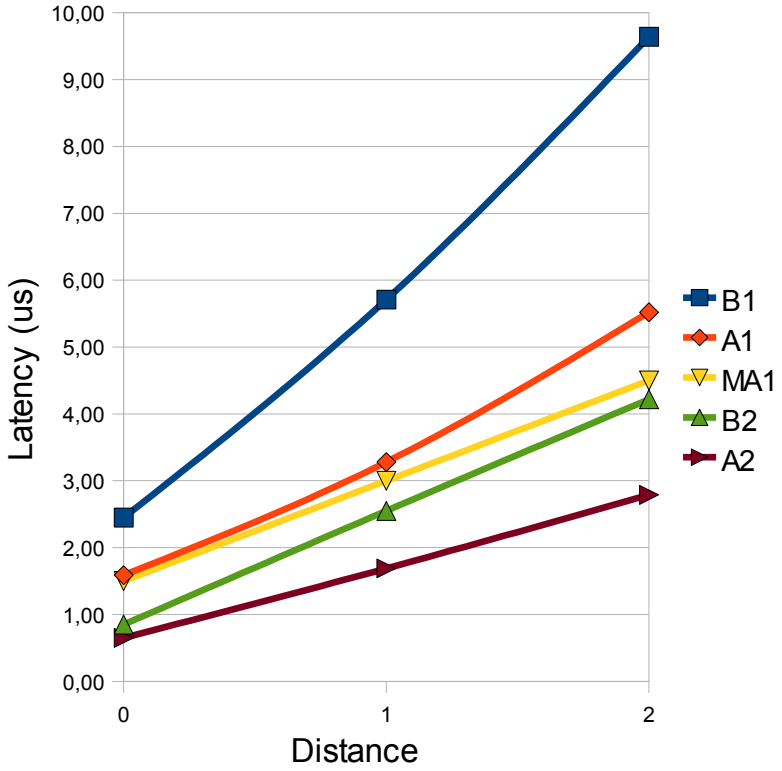


Latency of a message exchange

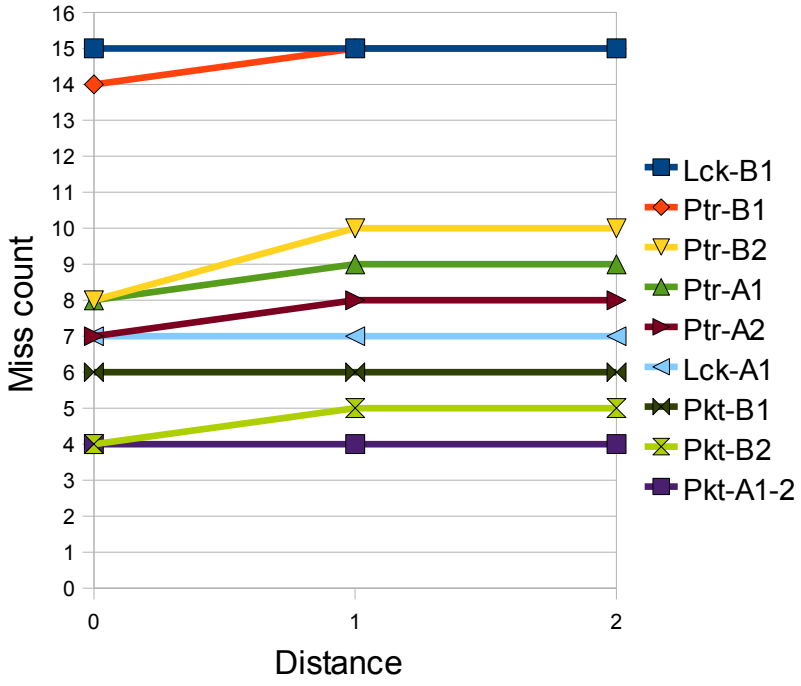
Number of cache misses

Evaluation results in several configurations

- Mapping of processes on the architecture → distance.
- Implementation of primitives (1, 2)
- Cache coherence protocol variant: A, B.



Latency of a message exchange



Number of cache misses

Conclusion

- Inexpensive modeling and evaluation:
  - Abstract model of a complex system.
  - About 2500 lines for modeling and verification.
  - LTS size: ~1,5M states. Minimized stochastic LTS: ~4,5K states.
  - A few minutes execution time.
- Yet, it has the potential to compare and analyze benchmark behavior in different configurations.
- Future work:
  - Additional performance figures.
  - Other MPI primitives.
  - Automatic production of LOTOS code.

Thank you