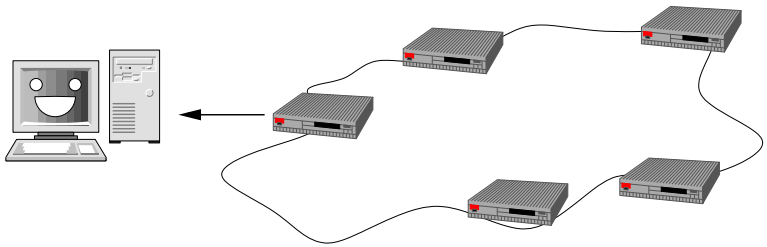
Phase-type Distributions for Realistic Modelling in Discrete-Event Simulation

Philipp Reinecke and Gábor Horváth

philipp.reinecke@fu-berlin.de
hgabor@webspn.hit.bme.hu

Motivation: The Restart Method



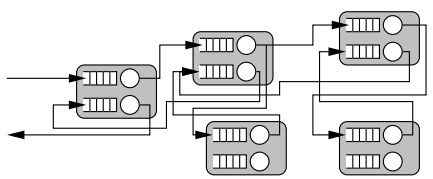
- Restart: A client sends a request. If there is no response within a reasonable time, the request is repeated
- Restart may reduce response-times
- Question: When should the client restart the request?
  - Small timeout $\rightarrow$ Low response-times, but also high additional system load
  - Large timeout $\rightarrow$ Low additional load, but high response-times
- Application scenarios: Service-Oriented Systems (SOAs), WMNs, etc.
- What happens if everyone does it?

Evaluation Approaches

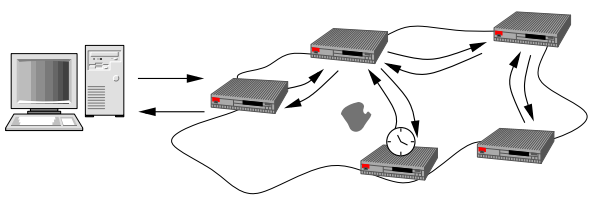
Analysis

$$F(x) = \int_0^x f(u)du$$

Simulation



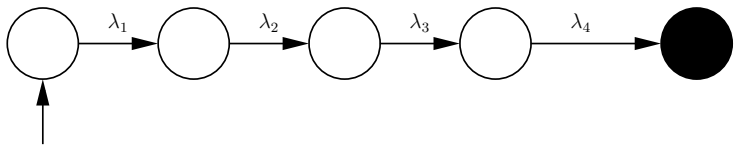
Experimental



Combined Approach

- Abstract methods give general results, but are often not realistic
- Practical methods are more realistic, but give less general results
- $\longrightarrow$ Combine methods to obtain realistic and general results
- Requirements:
  - Phenomena (e.g. response-times) must be modelled
  - Models are required
  - . . . must be accurate
  - . . . must be fast
  - . . . must be suitable for all abstraction levels
- Ideal models: Phase-type (PH) distributions.

Phase-type distributions



- A PH distribution is the distribution of the time to absorption in a Markov chain with one absorbing state
- Examples:
  - Exponential distribution
  - Hyperexponential distribution
  - Erlang distribution
  - Hypoexponential distribution

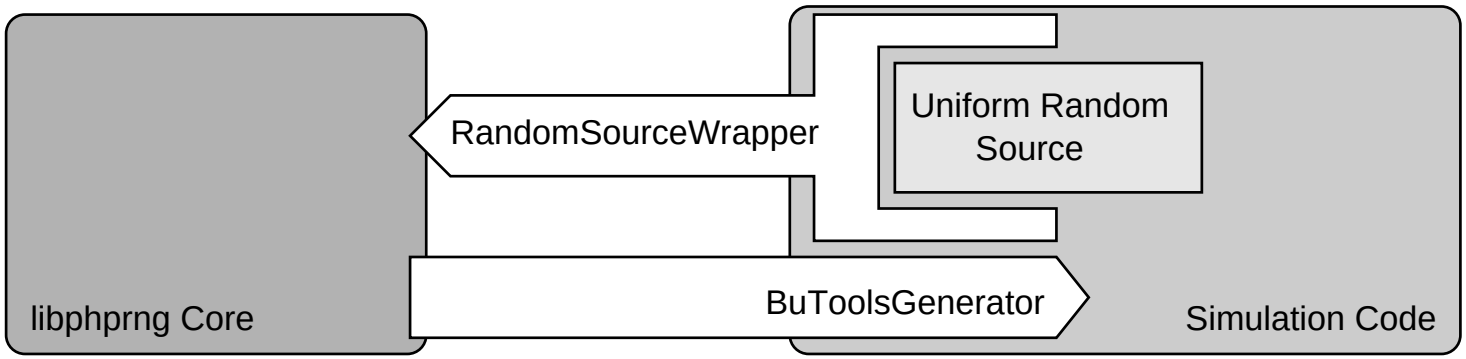
PH-Distributions for Modelling

- Use PH distributions to model delays, response-times, failure-times, etc. in test-beds, simulations, and abstract models
- Advantages over other distributions:
  - Flexibility $\rightarrow$ Capture important system properties by fitting PH distributions to measurements
  - Generic representations $\rightarrow$ Catch-all routines for random-variate generation
  - Markovian representations $\rightarrow$ Suitable for analytical approaches
- Seldom used in simulation
  - little-known
  - difficult theory
  - little to no support in simulators
  - efficiency concerns

The Libphprng Library

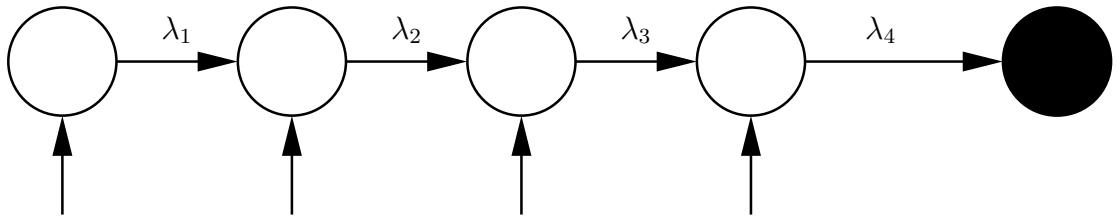
- A library for generating random variates from PH distributions
- Part of the Butools collection
  `http://webspn.hit.bme.hu/~butools`
- Advantages:
  - easy to use
  - portable between simulators
  - fast

Libphprng features



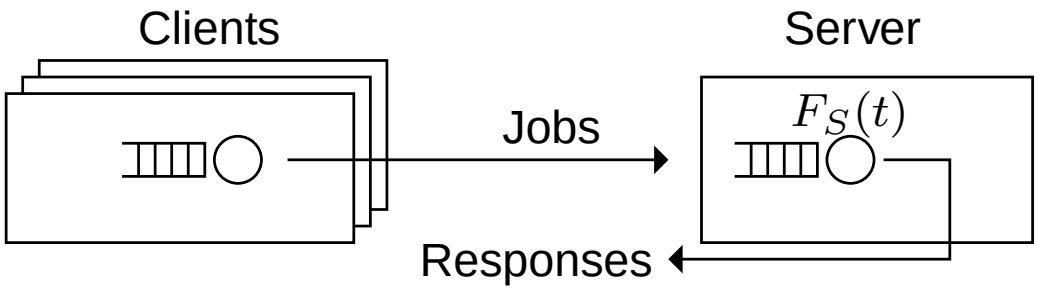
- Shared library with small wrapper code for the uniform random number stream
- Application:
  1. Create `BuToolsGenerator` object for the distribution
  2. Register uniform random number stream
  3. Draw random variates
- For other simulators: Write your own wrapper

Efficiency concerns



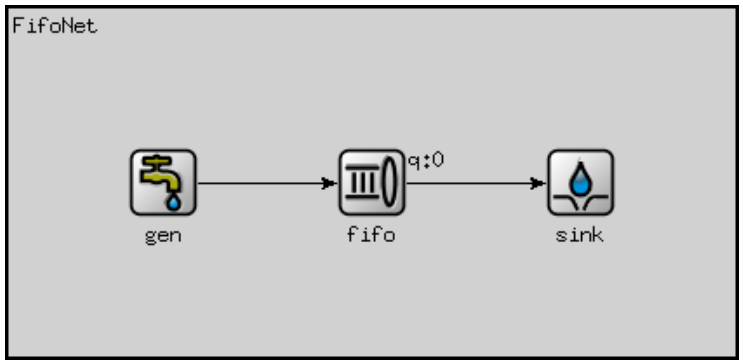
- Random-variate generation by 'playing' the Markov chain
- Costs depend on the structure and the algorithm . . . e.g. for a chain we do not need to randomly select the next state
- Structures are not unique
- Costs can be optimised by changing the structure
- Libphprng implements efficient algorithms and optimises the structure for random-variate generation

Evaluation



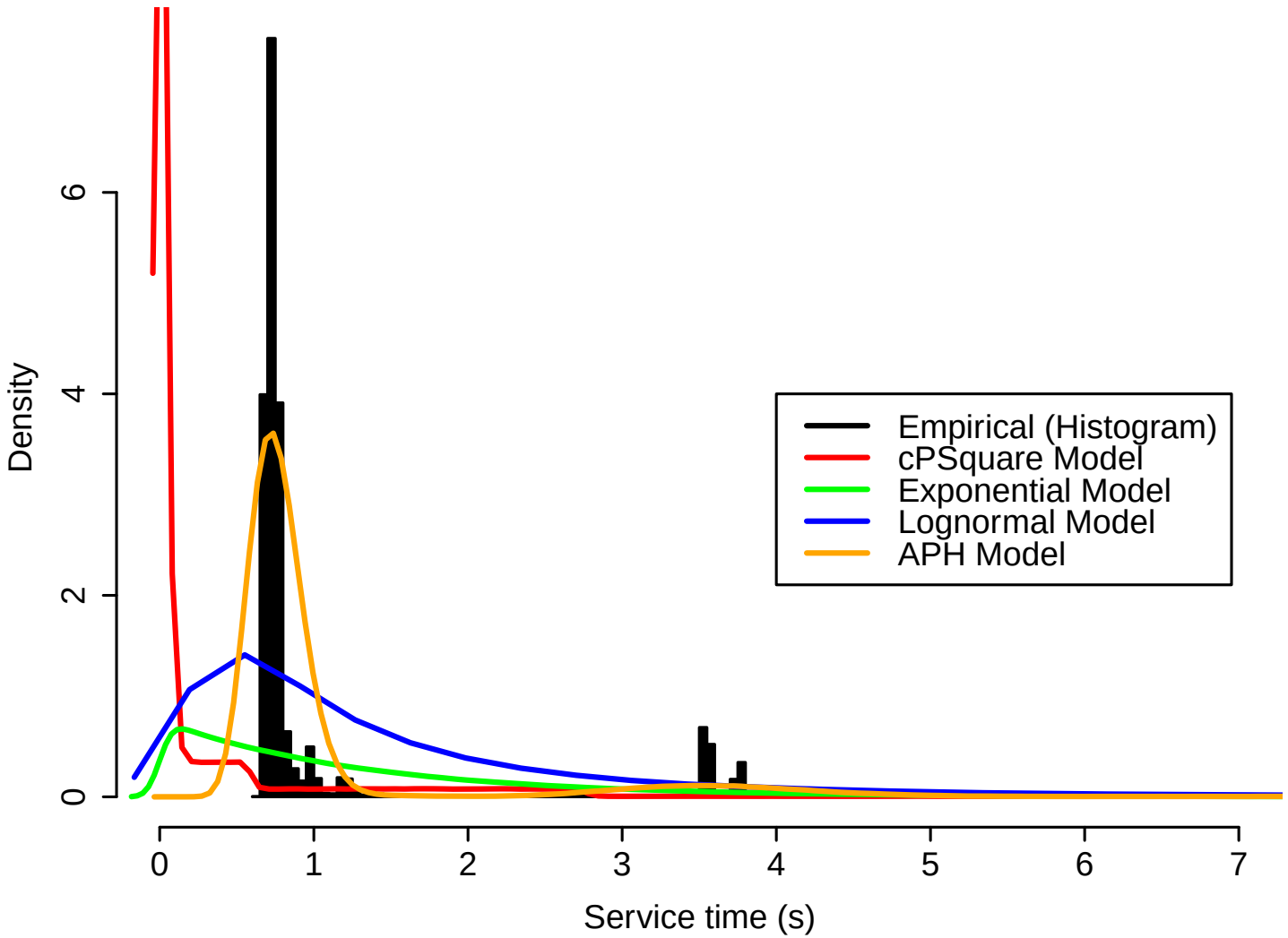
- Evaluation of quality and performance
- Quality: Evaluation of restart timeouts
- Different models:
  - `cPSquare`
  - Exponential distribution
  - Lognormal distribution
  - Phase-type distribution (50 phases)

Evaluation

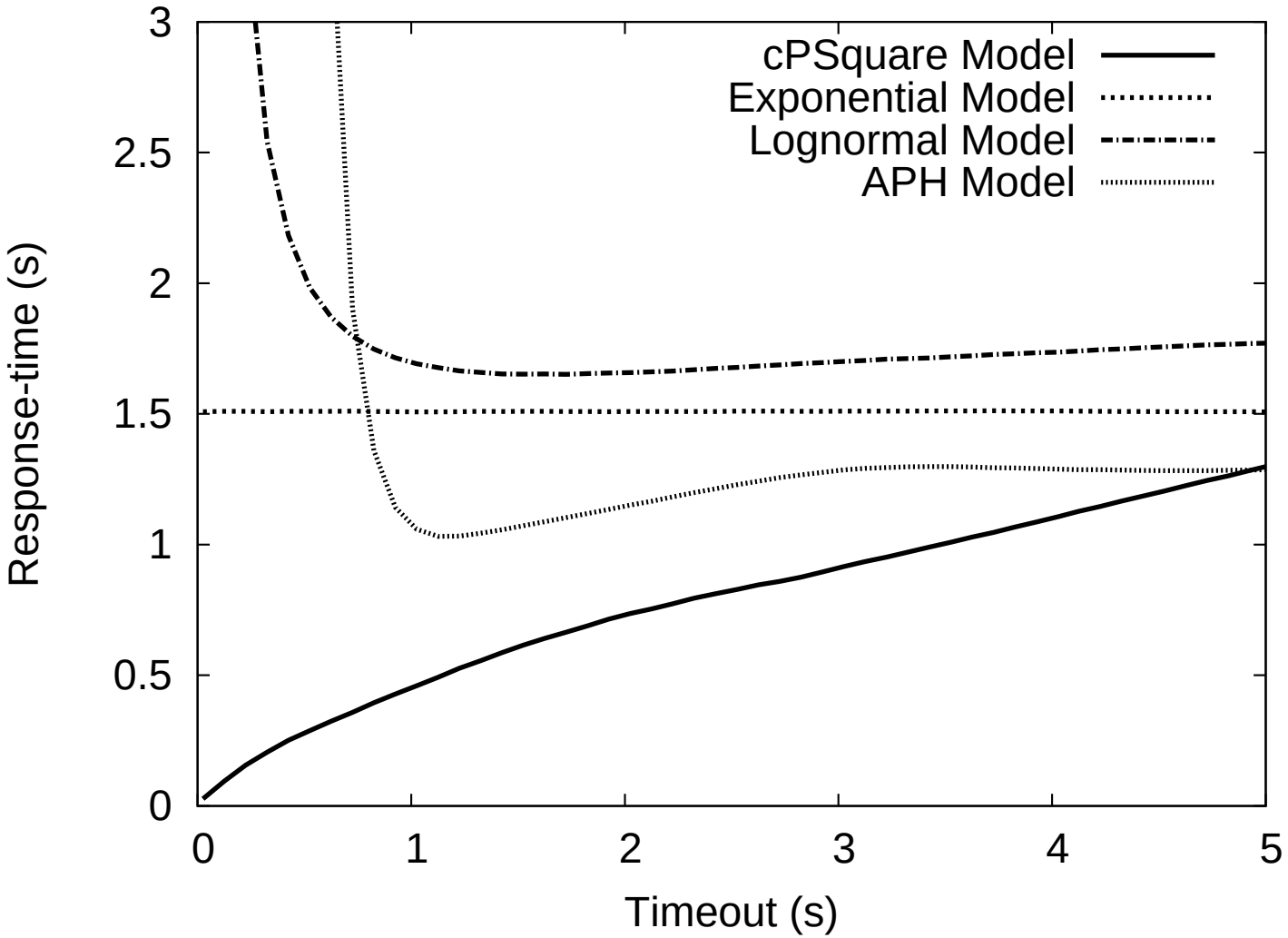


- Evaluation of quality and performance
- Quality: Evaluation of restart timeouts
- Different models:
  - cPSquare
  - Exponential distribution
  - Lognormal distribution
  - Phase-type distribution (50 phases)
- Performance: Simple source/sink model

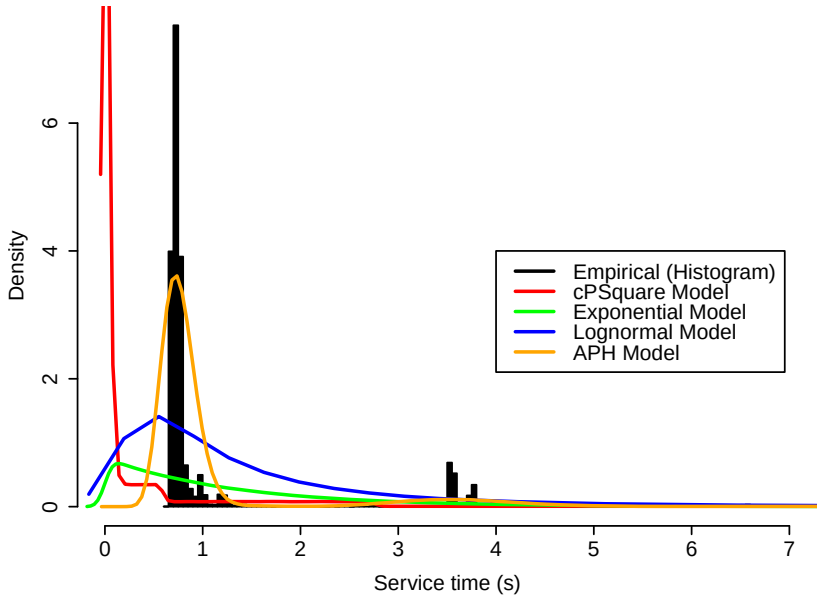
Evaluation: Quality

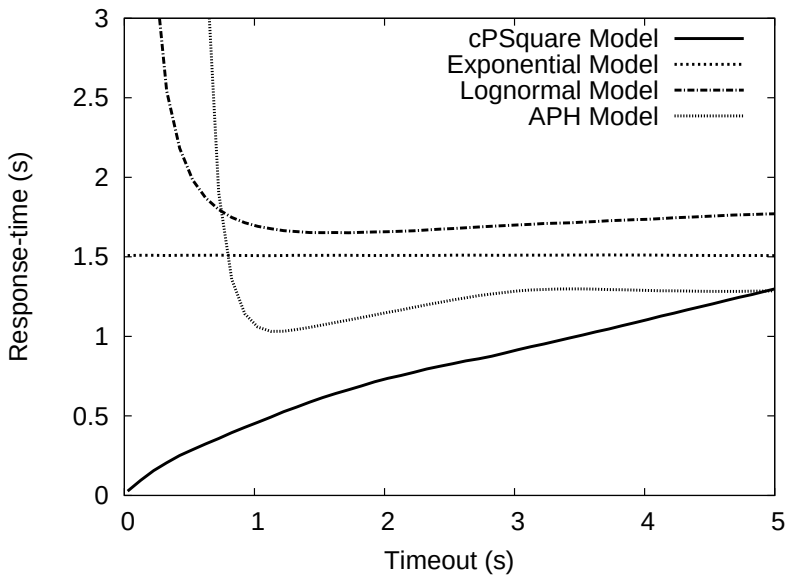


Evaluation: Quality



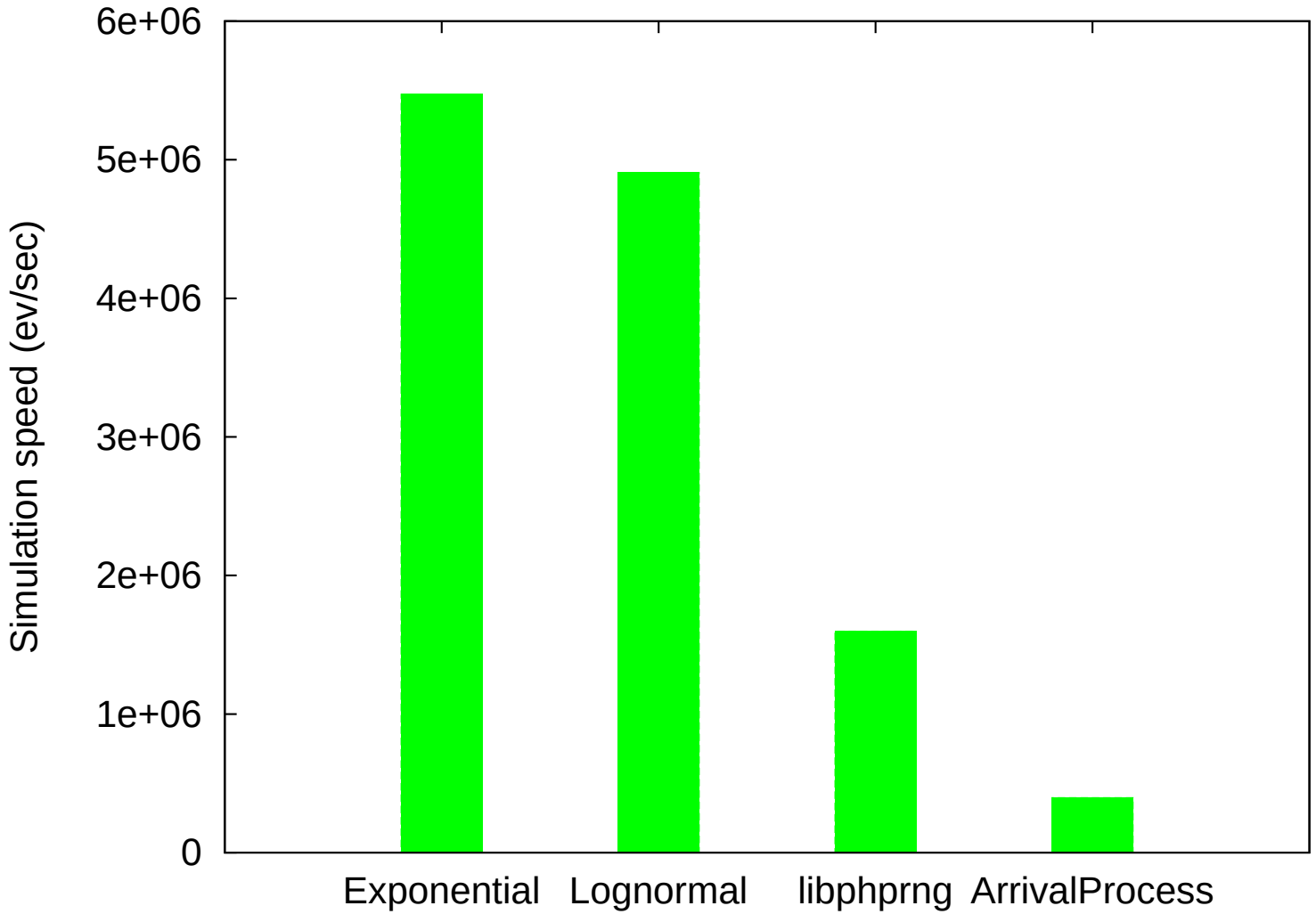
Evaluation: Quality



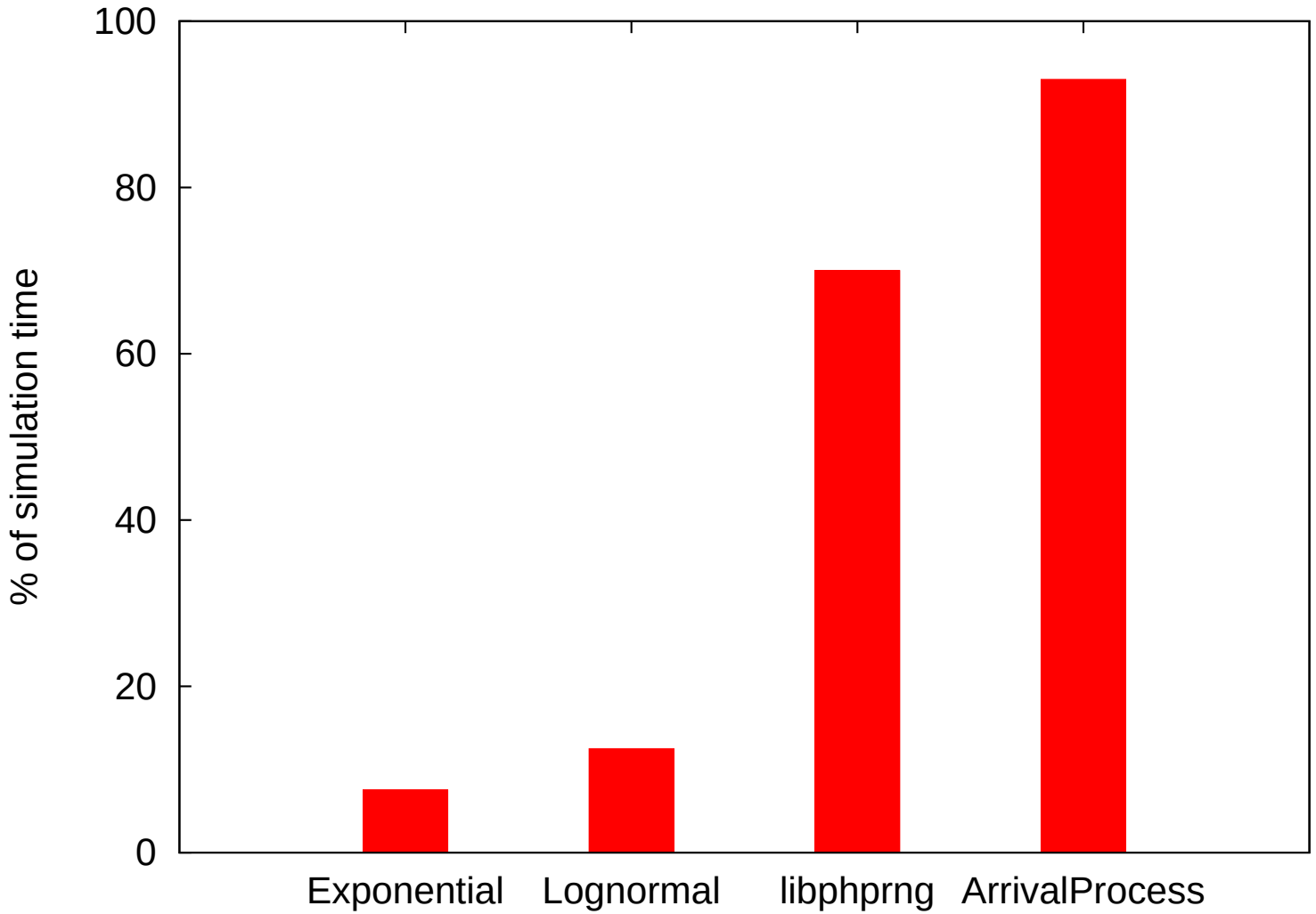


- Not all models capture the density well
- Comparison of results: Only the PH model shows the existence of an optimal timeout

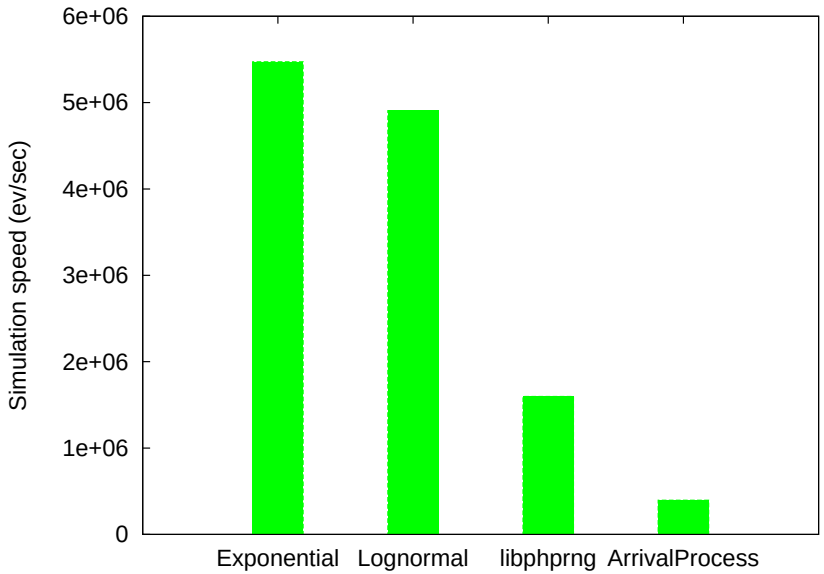
Evaluation: Performance

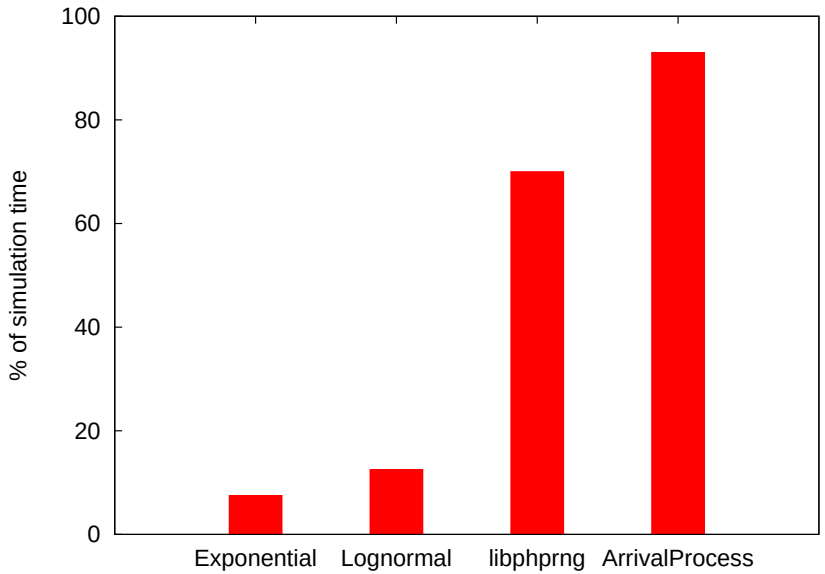


Evaluation: Performance



Evaluation: Performance





- Libphprng is less efficient than the simpler models
- Libphprng is more efficient than `ArrivalProcess` by Kriege et al. (2011) . . . but only supports PH

Conclusion

- Libphprng enables accurate and efficient modelling of distributions in simulations using PH distributions
- Libphprng is portable between simulators
- Available from

  `http://webspn.hit.bme.hu/~butools`

fin.