Synchronisation in PEPA models

Jeremy Bradley Stephen Gilmore Nigel Thomas

Email: jb@doc.ic.ac.uk nigel.thomas@ncl.ac.uk

stephen.gilmore@ed.ac.uk

Department of Computing, Imperial College London

LFCS, University of Edinburgh

Department of Computer Science, University of Newcastle

Produced with prosper and LaTeX

In stochastic models...

Synchronisation can significantly affect performance results

Presentation

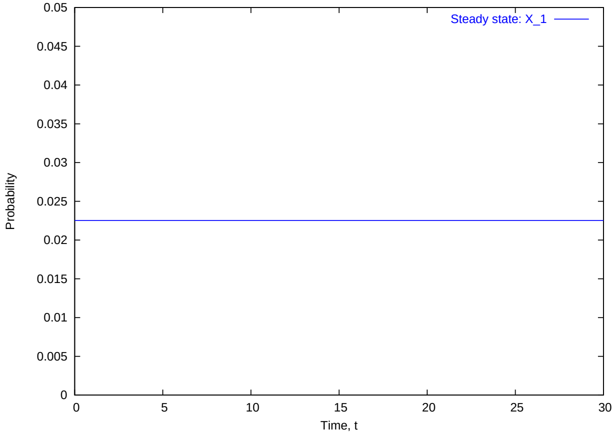
- PEPA and analysis
- The PEPA process algebra
- Synchronisation in practice
- Results

Types of Analysis

Steady-state and transient analysis in PEPA:

$$
\begin{aligned}
A1 &\stackrel{def}{=} (start, r_1).A2 + (pause, r_2).A3 \\
A2 &\stackrel{def}{=} (run, r_3).A1 + (fail, r_4).A3 \\
A3 &\stackrel{def}{=} (recover, r_1).A1 \\
AA &\stackrel{def}{=} (run, \top).(alert, r_5).AA \\
Sys &\stackrel{def}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
$$

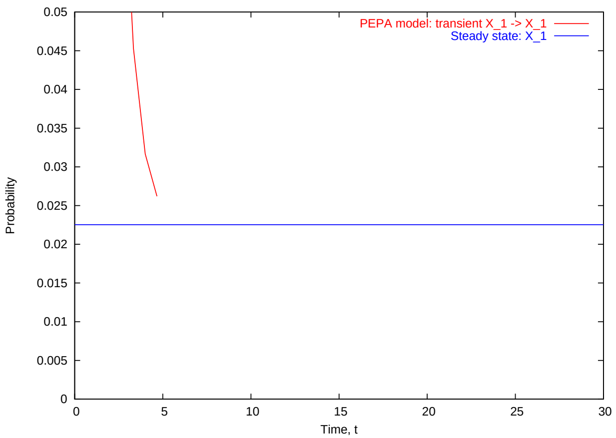
$\Rightarrow$



Types of Analysis

Steady-state and transient analysis in PEPA:

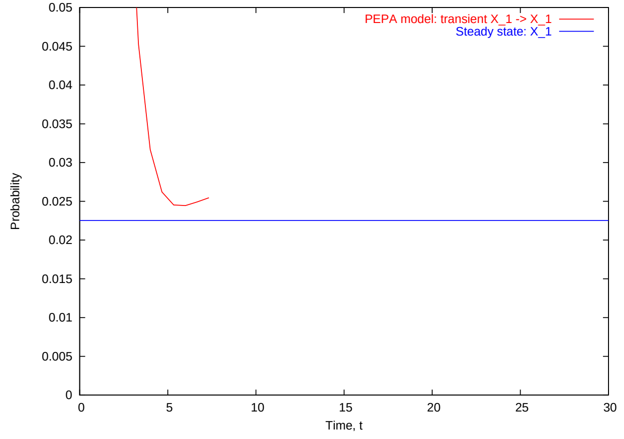
$$
\begin{array}{lcl}
A1 & \stackrel{\text{def}}{=} & (\text{start}, r_1).A2 + (\text{pause}, r_2).A3 \\
A2 & \stackrel{\text{def}}{=} & (\text{run}, r_3).A1 + (\text{fail}, r_4).A3 \\
A3 & \stackrel{\text{def}}{=} & (\text{recover}, r_1).A1 \\
AA & \stackrel{\text{def}}{=} & (\text{run}, \top).(\text{alert}, r_5).AA \\
Sys & \stackrel{\text{def}}{=} & AA \underset{\{run\}}{\bowtie} A1
\end{array}
\quad \Rightarrow
$$



Types of Analysis

Steady-state and transient analysis in PEPA:

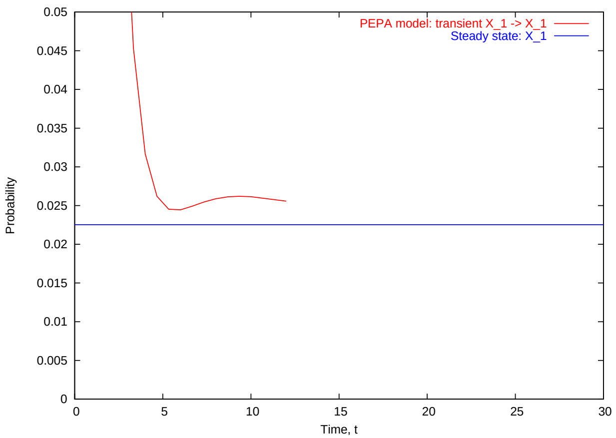
$$
\begin{array}{lcl}
A1 & \stackrel{\text{def}}{=} & (start, r_1).A2 + (pause, r_2).A3 \\
A2 & \stackrel{\text{def}}{=} & (run, r_3).A1 + (fail, r_4).A3 \\
A3 & \stackrel{\text{def}}{=} & (recover, r_1).A1 \\
AA & \stackrel{\text{def}}{=} & (run, \top).(alert, r_5).AA \\
Sys & \stackrel{\text{def}}{=} & AA \underset{\{run\}}{\bowtie} A1
\end{array}
\Rightarrow
$$



Types of Analysis

Steady-state and transient analysis in PEPA:

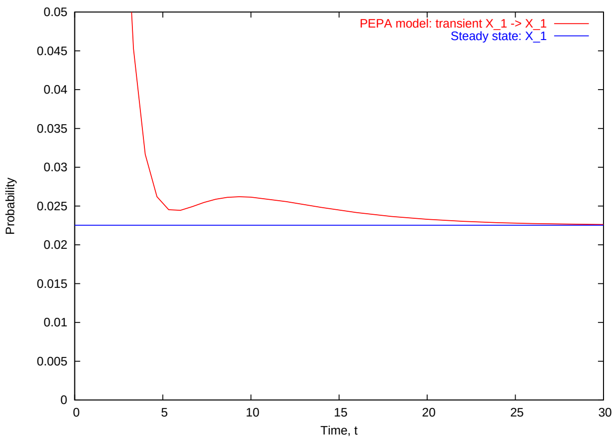
$$
\begin{aligned}
A1 &\stackrel{\text{def}}{=} (\text{start}, r_1).A2 + (\text{pause}, r_2).A3 \\
A2 &\stackrel{\text{def}}{=} (\text{run}, r_3).A1 + (\text{fail}, r_4).A3 \\
A3 &\stackrel{\text{def}}{=} (\text{recover}, r_1).A1 \\
AA &\stackrel{\text{def}}{=} (\text{run}, \top).(\text{alert}, r_5).AA \\
Sys &\stackrel{\text{def}}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
\quad \Rightarrow
$$



Types of Analysis

Steady-state and transient analysis in PEPA:

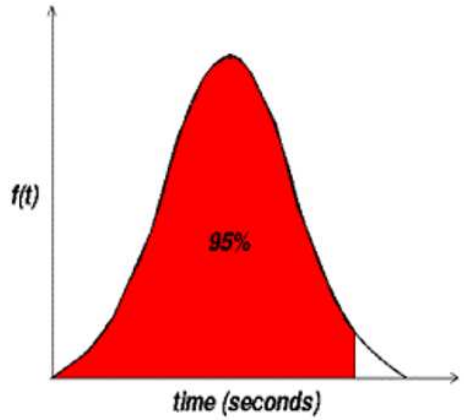
$$
\begin{aligned}
A1 &\stackrel{\text{def}}{=} (\text{start}, r_1).A2 + (\text{pause}, r_2).A3 \\
A2 &\stackrel{\text{def}}{=} (\text{run}, r_3).A1 + (\text{fail}, r_4).A3 \\
A3 &\stackrel{\text{def}}{=} (\text{recover}, r_1).A1 \\
AA &\stackrel{\text{def}}{=} (\text{run}, \top).(\text{alert}, r_5).AA \\
Sys &\stackrel{\text{def}}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
\quad \Rightarrow
$$



Passage-time Quantiles

Extract a passage-time density from a PEPA model:

$$
\begin{aligned}
A1 &\stackrel{\text{def}}{=} (\text{start}, r_1).A2 + (\text{pause}, r_2).A3 \\
A2 &\stackrel{\text{def}}{=} (\text{run}, r_3).A1 + (\text{fail}, r_4).A3 \\
A3 &\stackrel{\text{def}}{=} (\text{recover}, r_1).A1 \\
AA &\stackrel{\text{def}}{=} (\text{run}, \top).(\text{alert}, r_5).AA \\
Sys &\stackrel{\text{def}}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
\quad \Longrightarrow
$$



Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$
- Competitive choice: $P_1 + P_2$

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$
- Competitive choice: $P_1 + P_2$
- Cooperation: $P_1 \underset{L}{\bowtie} P_2$

Stochastic Process Algebra

PEPA syntax:

$$\mathrm{P} \;::=\; (\mathrm{a}, \lambda).\mathrm{P} \;\Big|\; \mathrm{P} + \mathrm{P} \;\Big|\; \mathrm{P} \underset{L}{\bowtie} \mathrm{P} \;\Big|\; \mathrm{P}/\mathrm{L} \;\Big|\; \mathrm{A}$$

- Action prefix: $(\mathrm{a}, \lambda).\mathrm{P}$
- Competitive choice: $\mathrm{P}_1 + \mathrm{P}_2$
- Cooperation: $\mathrm{P}_1 \underset{L}{\bowtie} \mathrm{P}_2$
- Action hiding: P/L

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$
- Competitive choice: $P_1 + P_2$
- Cooperation: $P_1 \underset{L}{\bowtie} P_2$
- Action hiding: P/L
- Constant label: A

State of the Art

- PEPA model: passage time/transient analysis - $O(10^8)$ states
- Semi-Markov PEPA: passage time/transient analysis - $O(10^7)$ states

PEPA: A Transmitter-Receiver

$$
\begin{aligned}
\text{System} &\stackrel{\text{def}}{=} (\text{Transmitter} \underset{\emptyset}{\bowtie} \text{Receiver}) \underset{\{\mathtt{transmit},\mathtt{receive}\}}{\bowtie} \text{Network} \\[2ex]
\text{Transmitter} &\stackrel{\text{def}}{=} (\text{transmit}, \lambda_1).(\text{t} - \text{recover}, \lambda_2).\text{Transmitter} \\
\text{Receiver} &\stackrel{\text{def}}{=} (\text{receive}, \top).(\text{r} - \text{recover}, \mu).\text{Receiver} \\
\text{Network} &\stackrel{\text{def}}{=} (\text{transmit}, \top).(\text{delay}, \nu_1).(\text{receive}, \nu_2).\text{Network}
\end{aligned}
$$

- A simple transmitter-receiver over a network

Apparent Rate

- Apparent rate of a component P is given by $r_a(\mathrm{P})$

Apparent Rate

- Apparent rate of a component P is given by $r_a(\mathrm{P})$
- Apparent rate describes the overall observed rate that P performs an a-action

Apparent Rate

- Apparent rate of a component P is given by $r_a(\mathrm{P})$
- Apparent rate describes the overall observed rate that P performs an a-action
- Apparent rate is given by:

$$r_a(\mathrm{P}) = \sum_{\mathrm{P} \xrightarrow{(a, \lambda_i)}} \lambda_i$$

Apparent Rate Examples

- $r_a(\mathrm{P} \xrightarrow{(\mathrm{a},\lambda)}) = \lambda$

Apparent Rate Examples

- $r_a(\mathrm{P} \xrightarrow{(\mathrm{a},\lambda)}) = \lambda$
- $r_a(\mathrm{P} \xrightarrow{(\mathrm{a},\top)}) = \top$

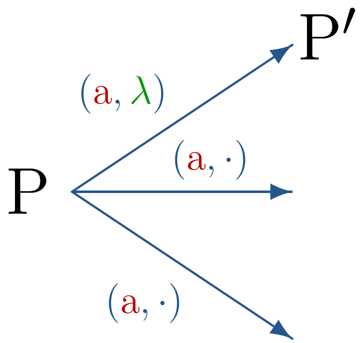
Apparent Rate Examples

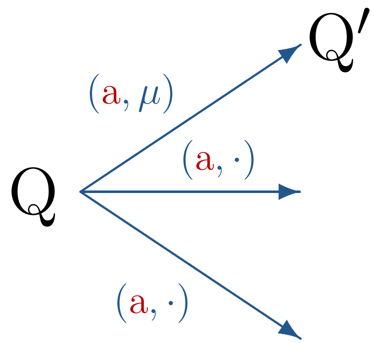
- $$r_a(\mathrm{P} \xrightarrow{(\mathrm{a},\lambda)}) = \lambda$$
- $$r_a(\mathrm{P} \xrightarrow{(\mathrm{a},\top)}) = \top$$
- $$r_a\left(\mathrm{P} \begin{array}{l} \nearrow^{(\mathrm{a},\lambda_1)} \\ \searrow_{(\mathrm{a},\lambda_2)} \end{array} \right) = \lambda_1 + \lambda_2$$

Apparent Rate Examples

- $r_a(P \xrightarrow{(a,\lambda)}) = \lambda$
- $r_a(P \xrightarrow{(a,\top)}) = \top$
- $$r_a\left(\begin{tikzcd}[column sep=huge] & {} \\ P \arrow[ur, "{(a,\lambda_1)}"] \arrow[dr, "{(a,\lambda_2)}"'] & \\ & {} \end{tikzcd} \right) = \lambda_1 + \lambda_2$$
- $$r_a\left(\begin{tikzcd}[column sep=huge] & {} \\ P \arrow[ur, "{(a,\top)}"] \arrow[dr, "{(a,\top)}"'] & \\ & {} \end{tikzcd} \right) = 2\top$$

Synchronisation Rate

- In PEPA, when synchronising two model components, P and Q where both P and Q enable many a-actions:
  - 

and


Synchronisation Rate

- In PEPA, when synchronising two model components, P and Q where both P and Q enable many a-actions:

  - $$\begin{tikzcd} & \mathrm{P}' \\ \mathrm{P} \arrow[ur, "(\mathrm{a},\lambda)"] \arrow[r, "(\mathrm{a},\cdot)"] \arrow[dr, "(\mathrm{a},\cdot)"'] & {} \\ & {} \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} & \mathrm{Q}' \\ \mathrm{Q} \arrow[ur, "(\mathrm{a},\mu)"] \arrow[r, "(\mathrm{a},\cdot)"] \arrow[dr, "(\mathrm{a},\cdot)"'] & {} \\ & {} \end{tikzcd}$$

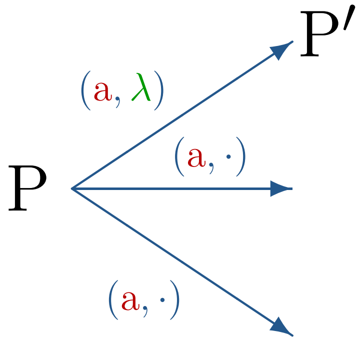
- The synchronised rate for $\mathrm{P} \underset{a}{\bowtie} \mathrm{Q} \xrightarrow{(\mathrm{a},R)} \mathrm{P}' \underset{a}{\bowtie} \mathrm{Q}'$ is:

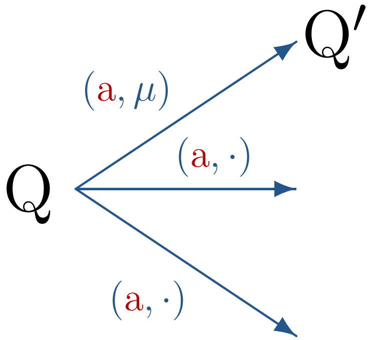
$$R = \frac{\lambda}{r_a(\mathrm{P})}\frac{\mu}{r_a(\mathrm{Q})}\min(r_a(\mathrm{P}), \mathrm{r_a}(\mathrm{Q}))$$

Approximate Synchronisation

- Some tools such as: Möbius, PRISM, PWB use an approximate synchronisation model

Approximate Synchronisation

- Some tools such as: Möbius, PRISM, PWB use an approximate synchronisation model
- With two model components, P and Q where both P and Q enable many a-actions:
  - 

    and
    

Approximate Synchronisation

- Some tools such as: Möbius, PRISM, PWB use an approximate synchronisation model
- With two model components, P and Q where both P and Q enable many a-actions:
  - $\begin{tikzcd} & & \mathrm{P}' \\ \mathrm{P} \arrow[urr, "(\mathrm{a}, \lambda)"] \arrow[rr, "(\mathrm{a}, \cdot)"] \arrow[drr, "(\mathrm{a}, \cdot)"'] & & {} \\ & & {} \end{tikzcd}$ and $\begin{tikzcd} & & \mathrm{Q}' \\ \mathrm{Q} \arrow[urr, "(\mathrm{a}, \mu)"] \arrow[rr, "(\mathrm{a}, \cdot)"] \arrow[drr, "(\mathrm{a}, \cdot)"'] & & {} \\ & & {} \end{tikzcd}$
- The *approximated* rate for $\mathrm{P} \underset{a}{\bowtie} \mathrm{Q} \xrightarrow{(\mathrm{a}, R)} \mathrm{P}' \underset{a}{\bowtie} \mathrm{Q}'$ is:

$$R = \min(\lambda, \mu)$$

Example

- As an example:
  - $\text{Client} \stackrel{\text{def}}{=} (\text{data}, \lambda).\text{Client}'$
  - $\text{Network} \stackrel{\text{def}}{=} (\text{data}, \top).\text{NetworkGo}$
    $+ (\text{data}, \top).\text{NetworkStall}$

Example

- As an example:
  - $\mathrm{Client} \stackrel{\mathrm{def}}{=} (\mathrm{data}, \lambda).\mathrm{Client}'$
  - $\mathrm{Network} \stackrel{\mathrm{def}}{=} (\mathrm{data}, \top).\mathrm{NetworkGo}$
    $+ (\mathrm{data}, \top).\mathrm{NetworkStall}$
- The combination $\mathrm{Client} \underset{data}{\bowtie} \mathrm{Network}$ should evolve with an overall data rate parameter of λ

Example

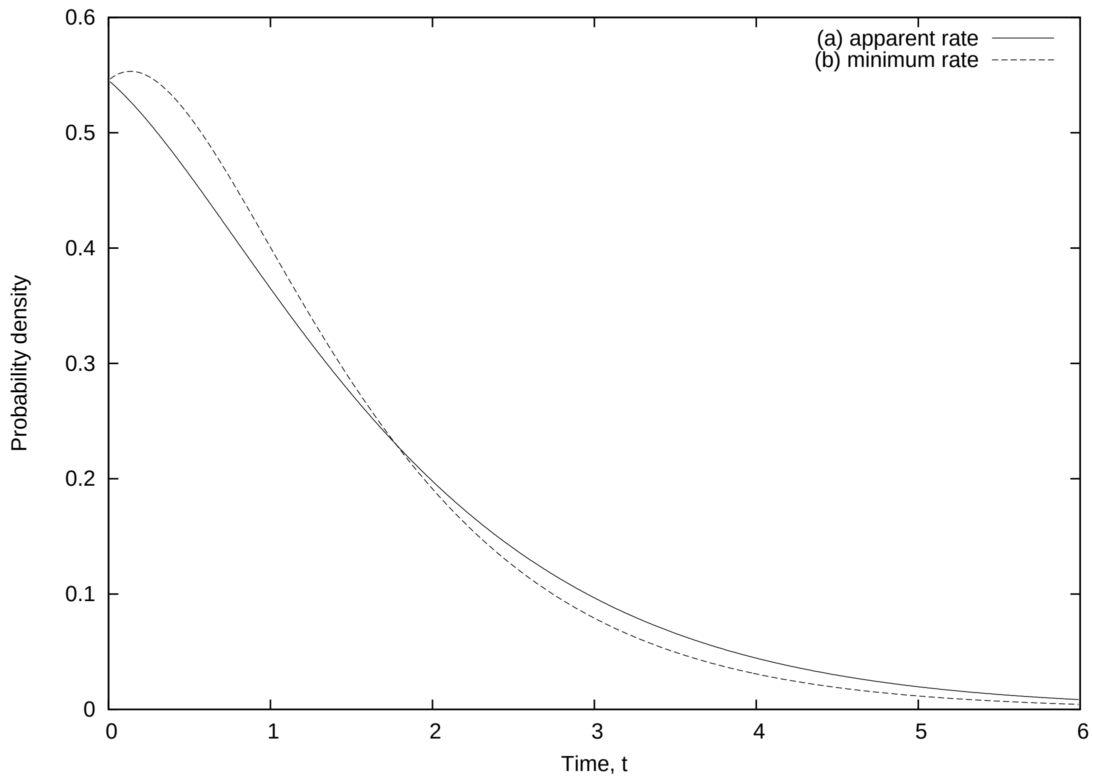
- As an example:
  - $\mathrm{Client} \stackrel{\mathrm{def}}{=} (\mathrm{data}, \lambda).\mathrm{Client}'$
  - $\mathrm{Network} \stackrel{\mathrm{def}}{=} (\mathrm{data}, \top).\mathrm{NetworkGo}$
    $+ \; (\mathrm{data}, \top).\mathrm{NetworkStall}$
- The combination $\mathrm{Client} \underset{data}{\bowtie} \mathrm{Network}$ should evolve with an overall data rate parameter of λ
- Under the tool approximation the overall synchronised rate becomes 2λ

Results: Multiple Passive

$$\begin{aligned} A &\stackrel{\text{def}}{=} (\text{run}, \lambda_1).(\text{stop}, \lambda_2).A \\ B &\stackrel{\text{def}}{=} (\text{run}, \top).(\text{pause}, \lambda_3).B \\ \text{Sys}_A &\stackrel{\text{def}}{=} A \underset{\{run\}}{\bowtie} (B \mid\mid B) \end{aligned}$$

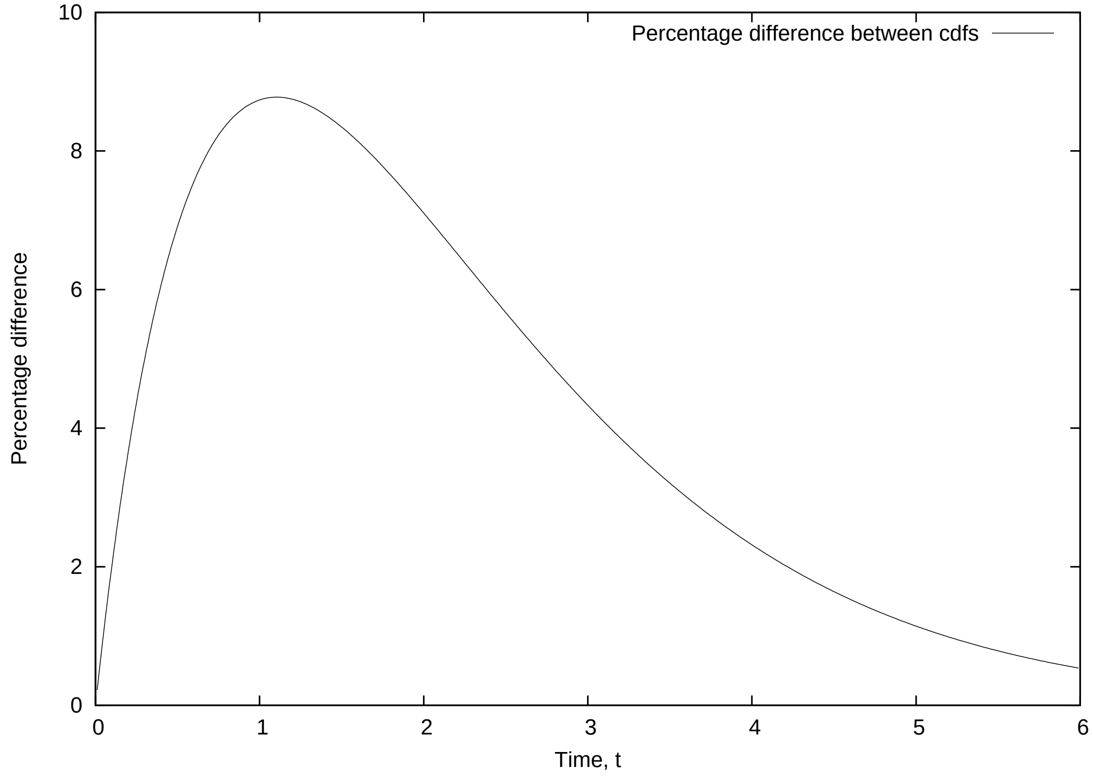
- Multiple passive ($\top$-rate) actions are enabled against a single real rate

Results: Multiple Passive



- Passage time density between consecutive stop actions

Results: Multiple Passive



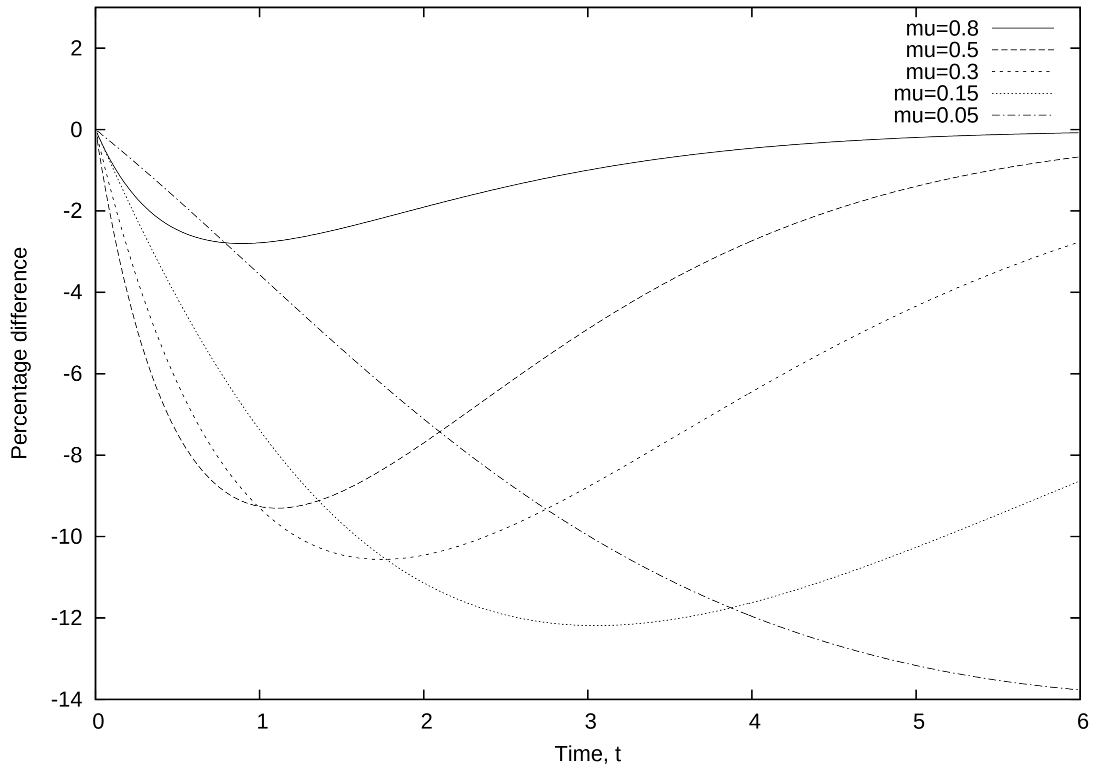
- Percentage difference in CDF functions over passage time between consecutive stop actions

Results: Multiple Active

$$\begin{aligned} \mathrm{A} &\stackrel{\text{def}}{=} (\mathrm{run}, \lambda_1).(\mathrm{stop}, \lambda_2).\mathrm{A} \\ \mathrm{B} &\stackrel{\text{def}}{=} (\mathrm{run}, \mu_1).(\mathrm{pause}, \lambda_3).\mathrm{B} \\ \mathrm{Sys_C} &\stackrel{\text{def}}{=} \mathrm{A} \underset{\{run\}}{\bowtie} (\mathrm{B} \,||\, \mathrm{B}) \end{aligned}$$

- Multiple real-rate actions (in $(\mathrm{B} \,||\, \mathrm{B})$) are synchronised against a single real-rate action (in A)

Results: Multiple Active



- Percentage difference in CDF functions over passage time between consecutive stop actions (for decreasing μ)

Isn't this really unusual?

- Q: How common is this kind of modelling problem? Isn't this bizarre non-determinism to see in a component?
- A: Having an explicit individual component with either:
  - $P \stackrel{\text{def}}{=} (a, \lambda).P' + (a, \mu).P''$ (multiple active)
  - $Q \stackrel{\text{def}}{=} (a, \top).Q' + (a, \top).Q''$ (multiple passive)
- ...might be unusal, but simple multi-agent synchronisation of $S \underset{\{a\}}{\bowtie} (R \mid\mid R \mid\mid \cdots \mid\mid R)$ for some S where $R \stackrel{\text{def}}{=} (a, \top).(b, \mu).R'$ causes just this problem
- This is a very common client–server architecture

Conclusion

- Synchronisation style makes a big difference to performance results!
- To summarise, using the tool approximation:
  - with multiple passive actions – sees an overestimation of passage-time results
  - with multiple active actions – sees an underestimation of passage-time results – why?