

“I am not a number: I am a free compiler”

Allan Clark and Stephen Gilmore

LFCS, University of Edinburgh

PASTA Workshop, London, June 30th 2006

Background: 1967

"The Prisoner"



"Number 6"

Background: 1967

"The Prisoner"



"Number 6"



"Puffball"

Today: 2006

“ipc”

- ▶ Exactly the same thing has now happened to
ipc: The Imperial PEPA Compiler
 - ▶ Just a number
 - ▶ Followed by puffball

Background

Quantitative Aspects of Services

The evaluation of a service level agreement (SLA) is a typical quantitative analysis problem.

Background

Quantitative Aspects of Services

The evaluation of a service level agreement (SLA) is a typical quantitative analysis problem.

Service level agreements

SLAs relate probabilities, paths and time.

“At least 90% of all requests receive a response within 3 seconds”

Background

Quantitative Aspects of Services

The evaluation of a service level agreement (SLA) is a typical quantitative analysis problem.

Service level agreements

SLAs relate **probabilities**, paths and time.

“At least **90%** of all requests receive a response within 3 seconds”

Background

Quantitative Aspects of Services

The evaluation of a service level agreement (SLA) is a typical quantitative analysis problem.

Service level agreements

SLAs relate probabilities, **paths** and time.

“At least 90% of all **requests receive a response** within 3 seconds”

Background

Quantitative Aspects of Services

The evaluation of a service level agreement (SLA) is a typical quantitative analysis problem.

Service level agreements

SLAs relate probabilities, paths and **time**.

“At least 90% of all requests receive a response within **3 seconds**”

SLAs in the early specification phase

- ▶ The difficulty at the early specification phase is that we have no measurement data which we can use to parameterise our high-level quantitative model (since the implementation has not yet been built),

SLAs in the early specification phase

- ▶ The difficulty at the early specification phase is that we have no measurement data which we can use to parameterise our high-level quantitative model (since the implementation has not yet been built),
- ▶ This uncertainty is manageable in practice because although we may not know precisely the value of the rate constants to be used in the model we may know **a range of values within which they will lie.**

SLAs in the early specification phase

- ▶ The difficulty at the early specification phase is that we have no measurement data which we can use to parameterise our high-level quantitative model (since the implementation has not yet been built),
- ▶ This uncertainty is manageable in practice because although we may not know precisely the value of the rate constants to be used in the model we may know **a range of values within which they will lie.**
- ▶ The problem then is simply to evaluate our model against our SLA measure a (possibly large) number of times.

Transient analysis and uniformisation

We work with a stochastic process algebra (PEPA) which has Continuous-Time Markov chains as the underlying mathematical model. Investigation of SLAs requires the transient analysis of a CTMC.

Transient analysis and uniformisation

We work with a stochastic process algebra (PEPA) which has Continuous-Time Markov chains as the underlying mathematical model. Investigation of SLAs requires the transient analysis of a CTMC.

Transient analysis

We are concerned with finding the transient state probability row vector $\pi(t) = [\pi_0(t), \dots, \pi_{n-1}(t)]$ where $\pi_i(t)$ denotes the probability that the CTMC is in state i at time t .

Transient analysis and uniformisation

We work with a stochastic process algebra (PEPA) which has Continuous-Time Markov chains as the underlying mathematical model. Investigation of SLAs requires the transient analysis of a CTMC.

Transient analysis

We are concerned with finding the transient state probability row vector $\pi(t) = [\pi_0(t), \dots, \pi_{n-1}(t)]$ where $\pi_i(t)$ denotes the probability that the CTMC is in state i at time t .

Transient and passage-time analysis of CTMCs proceeds by a numerical procedure called **uniformisation**.

Passage-time computation

Passage-time computation is concerned with knowing the probability of reaching a designated target state from a designated source state. It rests on two key sub-computations.

1. Finding the time to complete n hops ($n = 1, 2, 3, \dots$), which is an Erlang distribution with parameters n and q .
2. Finding the probability that the transition between source and target states occurs in exactly n hops.

Encoding service-level agreements in Continuous Stochastic Logic

A widely-used logic for model checking properties against continuous-time Markov chains is Continuous Stochastic Logic (CSL). The well-formed formulae of CSL are made up of *state formulae* ϕ and *path formulae* ψ .

$$\begin{aligned}\phi & ::= \text{true} \mid \text{false} \mid a \mid \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \mathcal{P}_{\bowtie p}[\psi] \mid \mathcal{S}_{\bowtie p}[\phi] \\ \psi & ::= X\phi \mid \phi U^I \phi \mid \phi U \phi\end{aligned}$$

where a is an atomic proposition, $\bowtie \in \{<, \leq, >, \geq\}$ is a relational parameter, $p \in [0, 1]$ is a probability, and I is an interval of \mathbb{R} . Derived logical operators such as implication (\Rightarrow) can be encoded in the usual way.

Case Study: Automotive crash scenario

The scenario under study considers the following sequence of events.

- ▶ A road traffic accident occurs. The car airbag deploys.
- ▶ Deployment of the air bag causes the on-board safety system to report the car's current location (obtained by GPS) to a pre-established accident report endpoint.
- ▶ The service at the reporting endpoint attempts to call the registered driver's mobile phone.
- ▶ If there is no answer to the call then medical assistance is dispatched to the reported location of the car (presuming that the driver has been incapacitated by injuries sustained in the accident).

PEPA model

$$Car_1 \stackrel{def}{=} (airbag, r_1).Car_2$$
$$Car_2 \stackrel{def}{=} (reportToService, r_2).Car_3$$
$$Car_3 \stackrel{def}{=} (processReport, r_3).Car_4$$

PEPA model

$$Car_1 \stackrel{def}{=} (airbag, r_1).Car_2$$
$$Car_2 \stackrel{def}{=} (reportToService, r_2).Car_3$$
$$Car_3 \stackrel{def}{=} (processReport, r_3).Car_4$$
$$Car_4 \stackrel{def}{=} (callDriversPhone, r_4).Car_5$$
$$Car_5 \stackrel{def}{=} (timeoutDriversPhone, r_5).Car_6$$

PEPA model

$$Car_1 \stackrel{def}{=} (airbag, r_1).Car_2$$
$$Car_2 \stackrel{def}{=} (reportToService, r_2).Car_3$$
$$Car_3 \stackrel{def}{=} (processReport, r_3).Car_4$$
$$Car_4 \stackrel{def}{=} (callDriversPhone, r_4).Car_5$$
$$Car_5 \stackrel{def}{=} (timeoutDriversPhone, r_5).Car_6$$
$$Car_6 \stackrel{def}{=} (rescue, r_6).Car_7$$
$$Car_7 \stackrel{def}{=} (awaitRescue, r_7).Car_1$$

Relation to model checking

Expressed as a CSL formula an example of the kind of question which we are asking is the following.

$$\mathit{airbag} \Rightarrow \mathcal{P}_{>0.9}[true \text{ U}^{[0,10]} rescue]$$

In words, this says “If the airbag in the car deploys, is it true with probability at least 0.9 that the rescue service will be sent within 10 minutes?”

Relation to model checking

We consider a more general form of the question which is the following

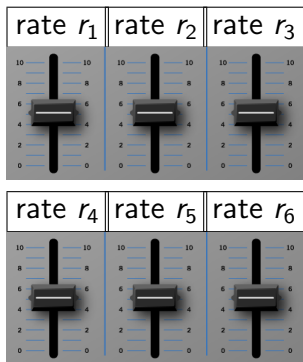
$$airbag \Rightarrow \mathcal{P}_{\bowtie p}[true \cup^{[0,10]} rescue]$$

We consider this for all relations $\bowtie \in \{<, \leq, >, \geq\}$ and for all values of the probability bound $0 \leq p \leq 1$. Further, we answer these general formulae not for only a single assignment of values to symbolic rate variables (as would be the case for conventional model checking) but across a range of assignments.

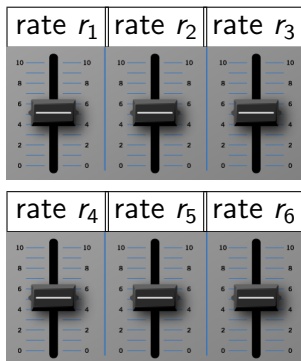
Rates constants and ranges

Rate	Value		Meaning
	min	max	
r_1	600.0	600.0	an airbag deploys in 1/10 of a second
r_2	2.0	10.0	the car can transmit location data in 6 to 30 seconds
r_3	0.5	1.5	it takes about one minute to register the incoming data
r_4	1.5	2.5	it takes about thirty seconds to call the driver's phone
r_5	1.0	60.0	give the driver from a second to one minute to answer
r_6	0.25	3.0	vary about one minute to decide to dispatch medical help
r_7	1.0	1.0	arbitrary value — the driver is now awaiting rescue

Experimentation

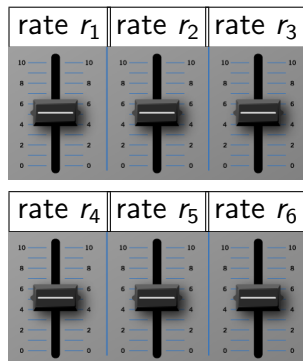


Experimentation



Say we discretize the values of five rates into 5 or 6 possible values.

Experimentation



Say we discretize the values of five rates into 5 or 6 possible values. $5 \times 5 \times 5 \times 5 \times 6 = 3750$ experiments.

Parameter sweep

Sometimes nothing succeeds like brute force

- ▶ Parameter sweep is an approach which falls into the class of problems commonly known as **embarrassingly parallelizable**.
 - ▶ many independent copies of the code are run
 - ▶ few of the complexities usually associated with parallel codes
- ▶ In this setting a simple approach based on a **Network of Workstations (NoW)** architecture will be effective in delivering the computational effort needed.

Parameter sweep

Sometimes nothing succeeds like brute force

- ▶ Parameter sweep is an approach which falls into the class of problems commonly known as **embarrassingly parallelizable**.
 - ▶ many independent copies of the code are run
 - ▶ few of the complexities usually associated with parallel codes
- ▶ In this setting a simple approach based on a **Network of Workstations (NoW)** architecture will be effective in delivering the computational effort needed.
- ▶ We used our Condor pool of 200 machines.
 - ▶ Some software development needed to make tools run on Condor.
 - ▶ Runtime varies depending on workload on machines.
 - ▶ Experiments completed in between 2 and 4 hours wall-clock time.

Workflow: ipc, Hydra and Uniform



ipc



hydra-s



uniform

ipc, Hydra and Uniform on Condor



ipc, Hydra and Uniform on Condor



ipc, Hydra and Uniform on Condor



ipc, Hydra and Uniform on Condor



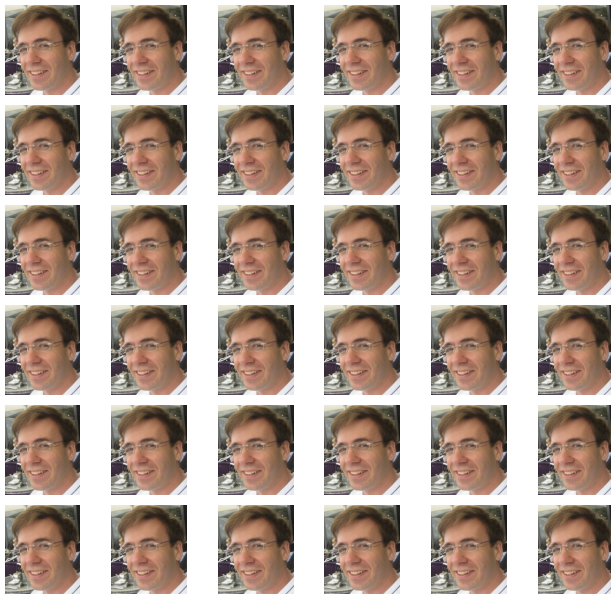
ipc, Hydra and Uniform on Condor



ipc, Hydra and Uniform on Condor



ipc, Hydra and Uniform on Condor



Generating Condor jobs

```
ipc --condor --permute --source=airbag --target=rescue \  
  --rates r5=1.0,16.0,31.0,46.0,61.0 \  
  --rates r2=2.0,4.0,6.0,8.0,10.0 \  
  --rates r3=0.5,0.75,1.0,1.25,1.5 \  
  --rates r4=1.5,1.75,2.0,2.25,2.5 \  
  --rates r6=0.25,0.75,1.25,1.75,2.25,2.75 \  
  --timestep 0.5 airbag.pepa
```

Running Condor jobs

```
ipc --processid 6 ...
```

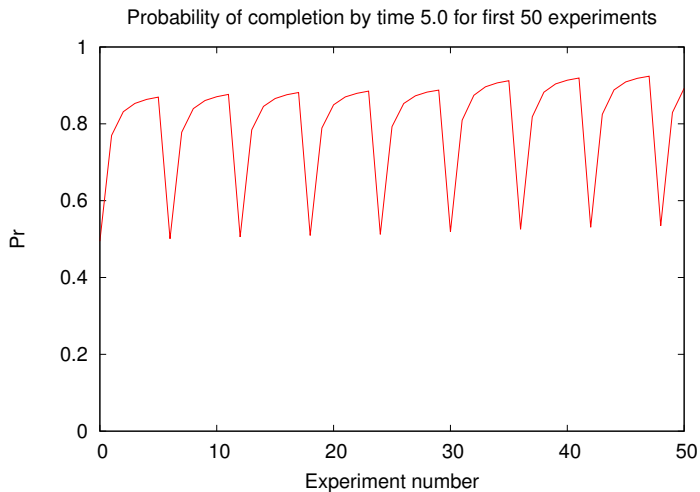
Being staunch

```
ipc --staunch ...
```

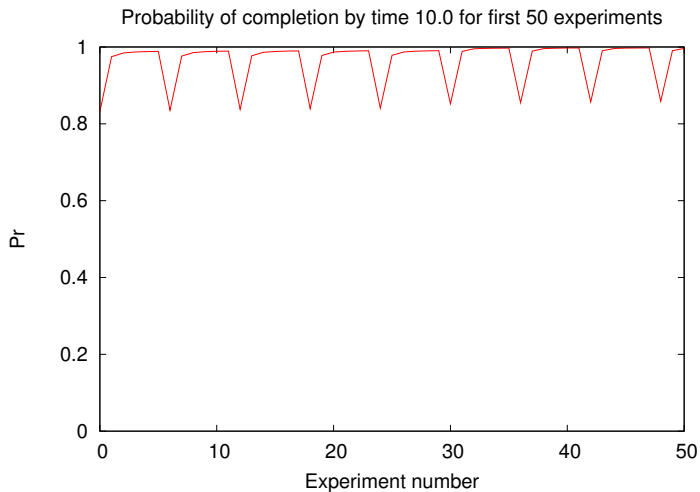
Puffball

- ▶ The results from the Condor run are post-processed by puffball.
- ▶ Projects views onto the results to obtain graphs and surface plots.

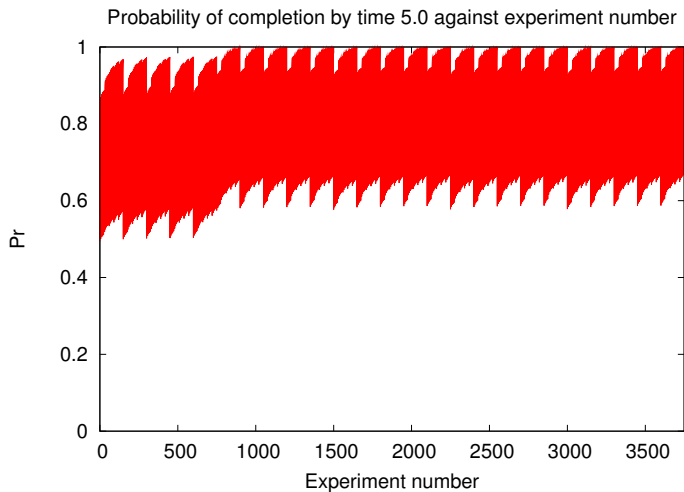
The first 50 experiments (five minute time bound)



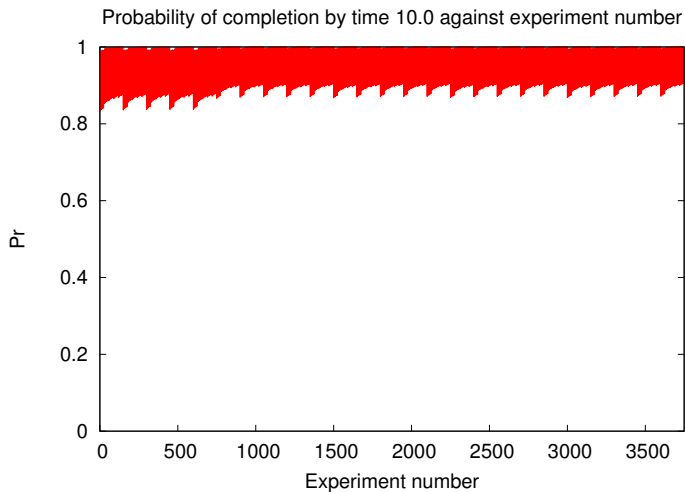
The first 50 experiments (ten minute time bound)



All 3750 experiments (five minute time bound)

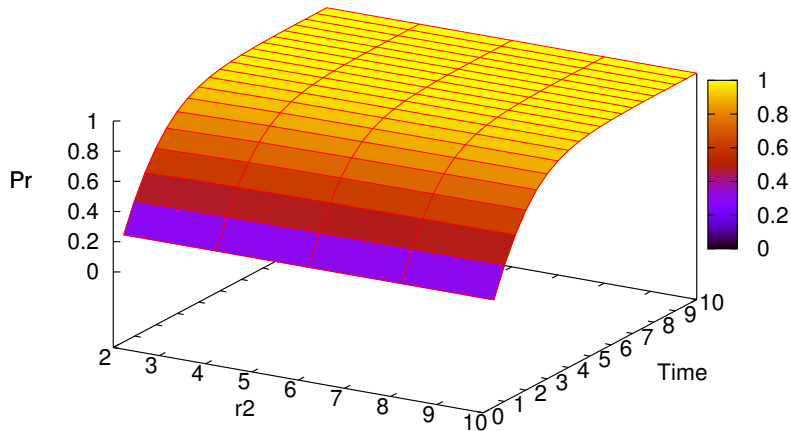


All 3750 experiments (ten minute time bound)



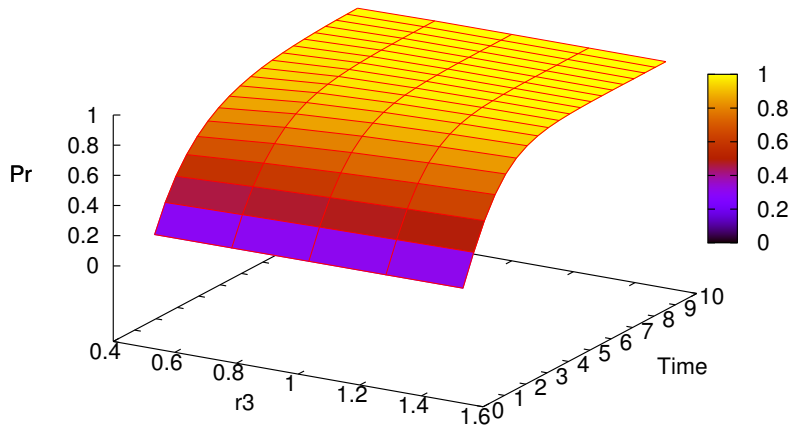
Sensitivity analysis on r_2

Sensitivity of cumulative distribution function to r_2



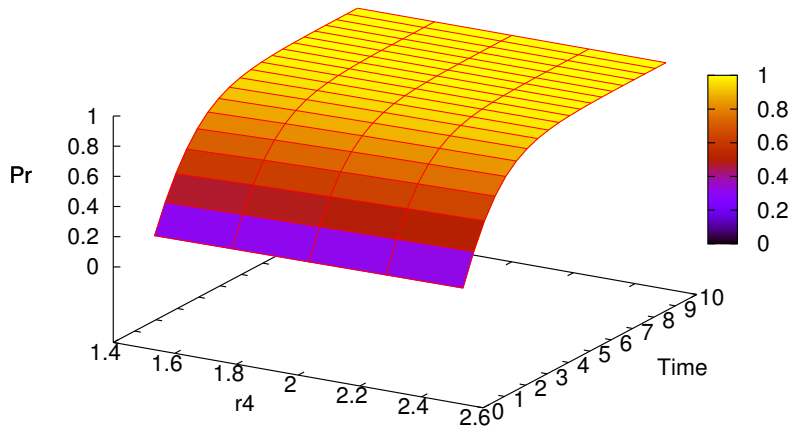
Sensitivity analysis on r_3

Sensitivity of cumulative distribution function to r_3



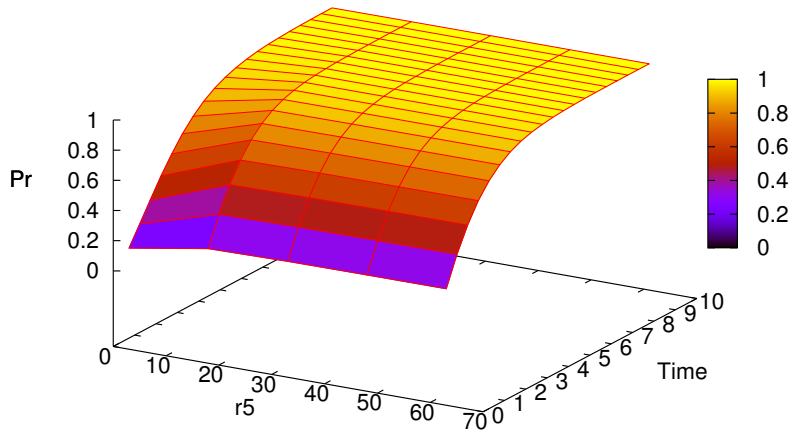
Sensitivity analysis on r_4

Sensitivity of cumulative distribution function to r_4



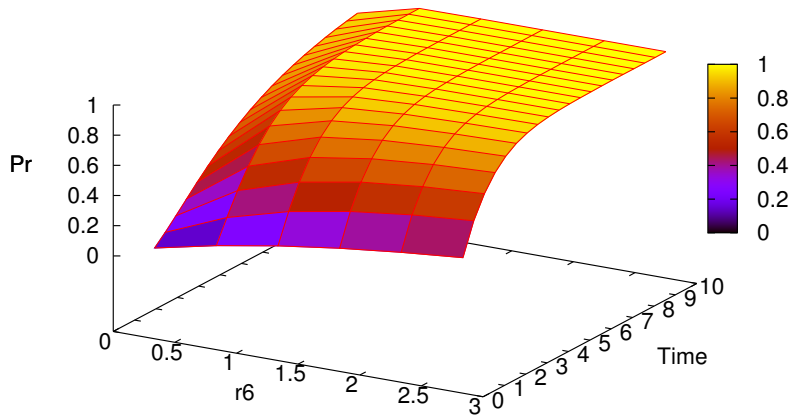
Sensitivity analysis on r_5

Sensitivity of cumulative distribution function to r_5



Sensitivity analysis on r_6

Sensitivity of cumulative distribution function to r_6



Conclusions

- ▶ Modified **ipc** to run on Condor as well as on a single machine.
- ▶ New tool for post-processing results: **puffball**
- ▶ Promising approach to do lots of analysis of PEPA models in a lightweight distributed way.

Conclusions

- ▶ Modified **ipc** to run on Condor as well as on a single machine.
- ▶ New tool for post-processing results: **puffball**
- ▶ Promising approach to do lots of analysis of PEPA models in a lightweight distributed way.
 - ▶ Today Edinburgh.
 - ▶ Tomorrow Edinburgh+Imperial+Newcastle+Glasgow+...

Acknowledgements

- ▶ The authors are supported by the SENSORIA project (EU FET-IST Global Computing 2 project 016004).
- ▶ We are grateful to Angelika Zobel and Nora Koch of F.A.S.T. München for the specification of the automotive case study.
- ▶ We modified the open-source software tool ipc developed by Jeremy Bradley and made freely available.
- ▶ We ran our models on the Condor cluster provided in the School of Informatics at Edinburgh and benefited from advice from Chris Cooke on using this effectively.

Web links

- ▶ <http://homepages.inf.ed.ac.uk/s9810217/software/ipc>
- ▶ <http://homepages.inf.ed.ac.uk/s9810217/software/puffball>