Process algebras for quantitative analysis: Lecture 5 — Service-level agreements

Stephen Gilmore

School of Informatics
The University of Edinburgh
Scotland

February 11, 2010

Quantitative analysis of critical systems

Summary of this talk

- We are reporting here on the quantitative analysis of a car assistance service currently in deployment which helps drivers who have been involved in an accident.
- We model the service in a *stochastic process algebra* PEPA which has Continuous-Time Markov Chains (CTMCs) as the underlying mathematical model.
- We evaluate a quantitative measure of the service using procedures of numerical linear algebra in order to determine whether or not the service meets its advertised *service level agreement*.

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 in general with SLAs at the early specification phase is that we have no measurement data which we can use to parameterise our high-level quantitative model.

SLAs in the early specification phase

- The difficulty in general with SLAs at the early specification phase is that we have no measurement data which we can use to parameterise our high-level quantitative model.
- 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 in general with SLAs at the early specification phase is that we have no measurement data which we can use to parameterise our high-level quantitative model.
- 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.

Markov chains

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

Markov chains

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

Markov chains

Markov chains are finite state stochastic processes. The transition system of a Markov chain can be stored as a generator matrix, where $Q_{ij} = r$ means that there is a transition from state i to state j at rate r.

Transient analysis

Transient analysis

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

Uniformisation

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

Uniformisation

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

Uniformisation

The generator matrix, Q, is "uniformized" with:

$$P = Q/q + I$$

where $q > \max_i |Q_{ii}|$. This process transforms a CTMC into one in which all states have the same mean holding time $1/q$.

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 ($n = 1, 2, 3, \ldots$), 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.

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).

Service-Level Agreement

The service level agreement related to this scenario concerns the response time of the passage from the deployment of the airbag to the dispatch of medical assistance.

Service-Level Agreement

The service level agreement related to this scenario concerns the response time of the passage from the deployment of the airbag to the dispatch of medical assistance.

The parameters of our modelling study include:

- the rate at which information on the location of the car—and any other pertinent information such as speed on impact, engine status, and other diagnostic information obtained from the on-board diagnostic systems and controllers—can be reported to the accident reporting service;

Service-Level Agreement

The service level agreement related to this scenario concerns the response time of the passage from the deployment of the airbag to the dispatch of medical assistance.

The parameters of our modelling study include:

- the rate at which information on the location of the car—and any other pertinent information such as speed on impact, engine status, and other diagnostic information obtained from the on-board diagnostic systems and controllers—can be reported to the accident reporting service;
- the time taken to confirm that the driver is not answering their mobile telephone; and

Service-Level Agreement

The service level agreement related to this scenario concerns the response time of the passage from the deployment of the airbag to the dispatch of medical assistance.

The parameters of our modelling study include:

- the rate at which information on the location of the car—and any other pertinent information such as speed on impact, engine status, and other diagnostic information obtained from the on-board diagnostic systems and controllers—can be reported to the accident reporting service;
- the time taken to confirm that the driver is not answering their mobile telephone; and
- the time taken to contact the emergency services to dispatch medical assistance.

Service-Level Agreement: uncertainty

None of the rate parameters are known exactly, but their average values are known to lie within a range of acceptable operation. We are, of course, interested in worst case bounds on passage-time quantiles and also in best case analysis but also in the variety of possible responses in between.

PEPA model — Car events

We consider the sequence of events which begins with the deployment of the airbag after the crash and finishes with the dispatch of the medical response team. The first phase of the sequence is concerned with relaying the information to the remote service, reporting the accident. When the diagnostic report from the car is received the service processes the report and matches it to the driver information stored on their database.

PEPA model — Car events

We consider the sequence of events which begins with the deployment of the airbag after the crash and finishes with the dispatch of the medical response team. The first phase of the sequence is concerned with relaying the information to the remote service, reporting the accident. When the diagnostic report from the car is received the service processes the report and matches it to the driver information stored on their database.

Car events

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

PEPA model — Service actions

The second phase of this passage through the system focuses on the attempted dialogue between the service and the registered driver of the car. We consider the case where the driver does not answer the incoming call because this is the case which leads to the medical response team being sent.

PEPA model — Service actions

The second phase of this passage through the system focuses on the attempted dialogue between the service and the registered driver of the car. We consider the case where the driver does not answer the incoming call because this is the case which leads to the medical response team being sent.

Service actions

$$
\begin{aligned}
Car_3 &\stackrel{def}{=} (processReport, r_3).Car_4 \\
Car_4 &\stackrel{def}{=} (callDriversPhone, r_4).Car_5 \\
Car_5 &\stackrel{def}{=} (timeoutDriversPhone, r_5).Car_6
\end{aligned}
$$

PEPA model — Outcomes

The service makes a final check on the execution of the procedure before the decision is taken to send medical help. At this stage the driver is awaiting rescue.

PEPA model — Outcomes

The service makes a final check on the execution of the procedure before the decision is taken to send medical help. At this stage the driver is awaiting rescue.

Outcomes

$$Car_6 \stackrel{def}{=} (rescue, r_6).Car_7$$
$$Car_7 \stackrel{def}{=} (awaitRescue, r_7).Car_1$$

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).

$$
\begin{array}{lrcl}
(\textit{state formulae}) & \phi & ::= & \textit{true} \mid \textit{false} \mid a \mid \\
 & & & \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \\
 & & & \mathcal{P}_{\bowtie p}[\psi] \mid \mathcal{S}_{\bowtie p}[\phi] \\
(\textit{path formulae}) & \psi & ::= & \mathrm{X}\,\phi \mid \phi\ \mathrm{U}^{I}\ \phi \mid \phi\ \mathrm{U}\ \phi
\end{array}
$$

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 $\mathbb{R}$.

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).

$$
\begin{array}{lrcl}
(\textit{state formulae}) & \phi & ::= & \textit{true} \mid \textit{false} \mid a \mid \\
 & & & \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \\
 & & & \mathcal{P}_{\bowtie p}[\psi] \mid \mathcal{S}_{\bowtie p}[\phi] \\
(\textit{path formulae}) & \psi & ::= & \mathrm{X}\,\phi \mid \phi\ \mathrm{U}^{I}\ \phi \mid \phi\ \mathrm{U}\ \phi
\end{array}
$$

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 $\mathbb{R}$.

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).

$$
\begin{array}{lrcl}
(\textit{state formulae}) & \phi & ::= & \textit{true} \mid \textit{false} \mid a \mid \\
 & & & \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \\
 & & & \mathcal{P}_{\bowtie p}[\psi] \mid \mathcal{S}_{\bowtie p}[\phi] \\
(\textit{path formulae}) & \psi & ::= & \mathrm{X}\,\phi \mid \phi\ \mathrm{U}^{I}\ \phi \mid \phi\ \mathrm{U}\ \phi
\end{array}
$$

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 $\mathbb{R}$.

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).

$$
\begin{array}{lrcl}
(\textit{state formulae}) & \phi & ::= & \textit{true} \mid \textit{false} \mid a \mid \\
 & & & \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \\
 & & & \mathcal{P}_{\bowtie p}[\psi] \mid \mathcal{S}_{\bowtie p}[\phi] \\
(\textit{path formulae}) & \psi & ::= & \mathrm{X}\,\phi \mid \phi\ \mathrm{U}^{I}\ \phi \mid \phi\ \mathrm{U}\ \phi
\end{array}
$$

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 $\mathbb{R}$.

Relation to model checking

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

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

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

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

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

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

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

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

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

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

We consider this for all relations $\bowtie \in \{ <, \leq, >, \geq \}$ and for all values of the probability bound $0 \leq p \leq 1$.

Relation to model checking

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

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

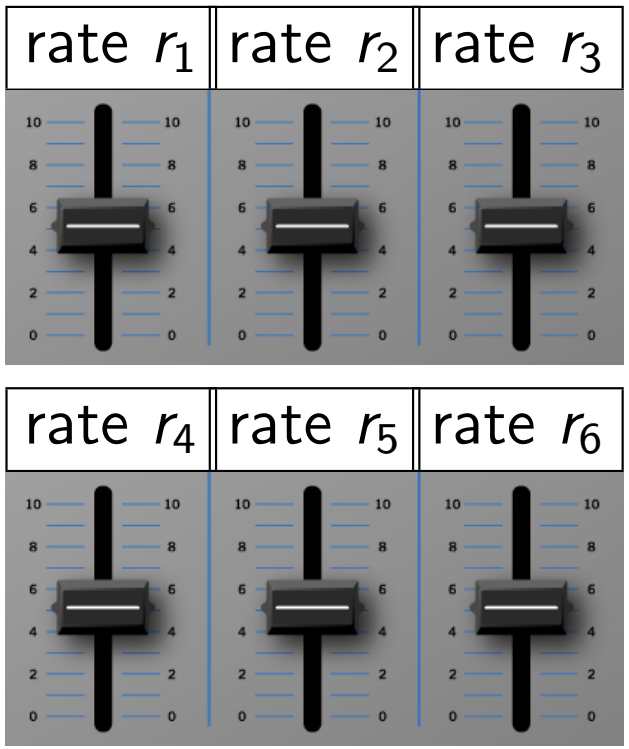
We consider this for all relations $\bowtie \in \{ <, \leq, >, \geq \}$ and for all values of the probability bound $0 \leq p \leq 1$.

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.

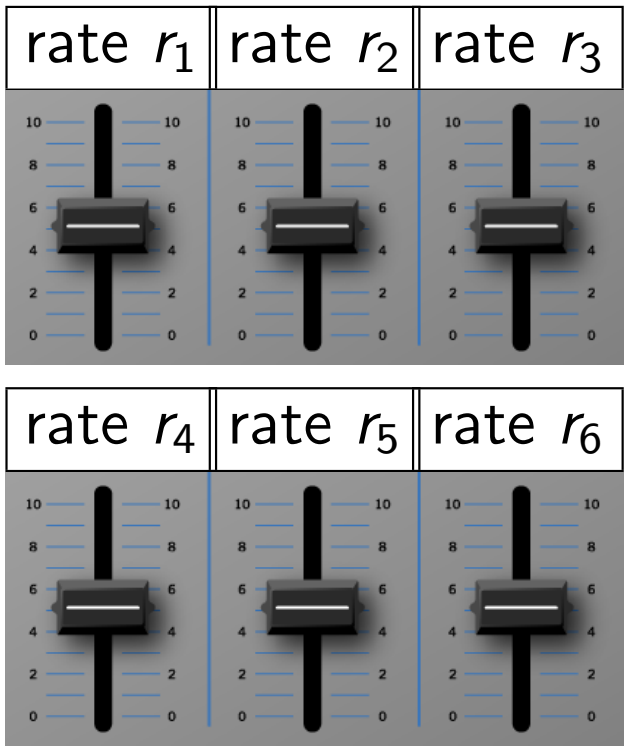
Rates constants and ranges

Rate	Value min	Value max	Meaning
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

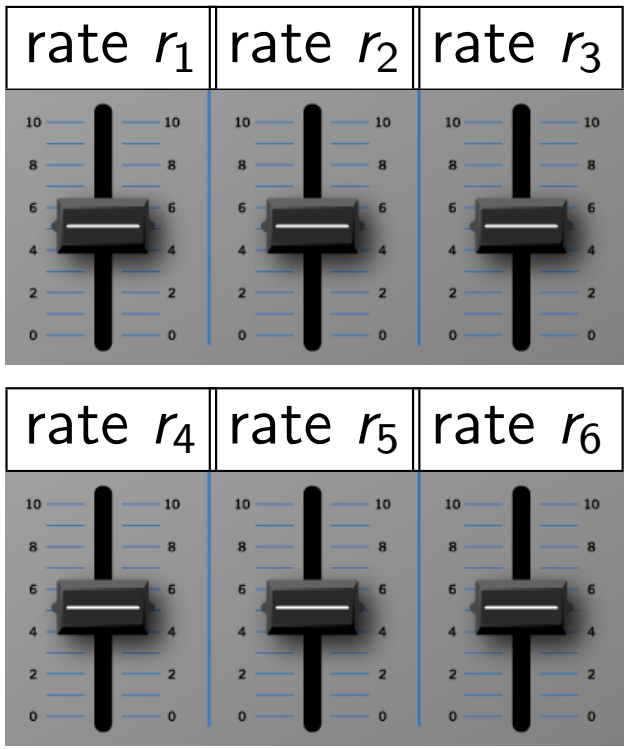


Experimentation



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

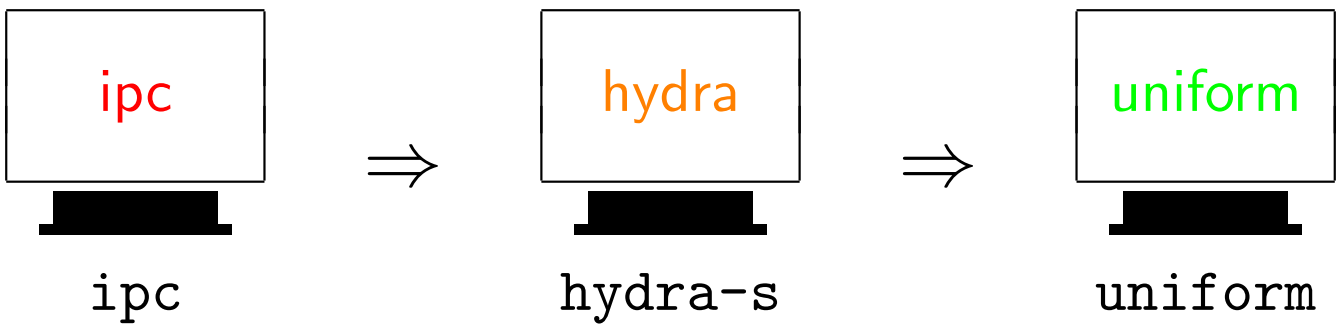
- Parameter sweep is an approach which falls into the class of problems commonly known as embarrassingly parallelizable.
  - many independent copies of the code are run
  - few of the complexities usually associated with parallel codes
- In this setting a simple approach based on a Network of Workstations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

Parameter sweep

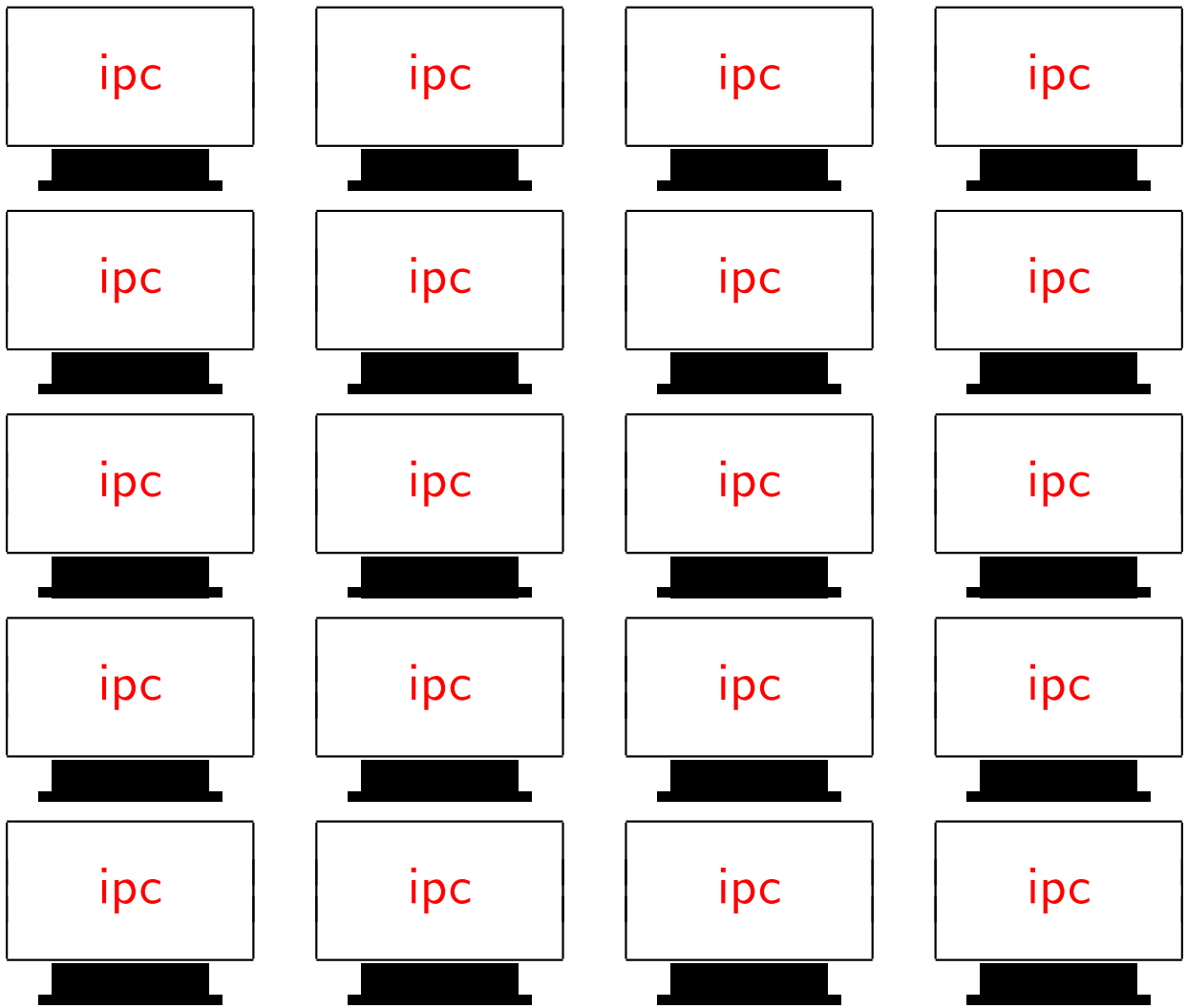
Sometimes nothing succeeds like brute force

- 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.

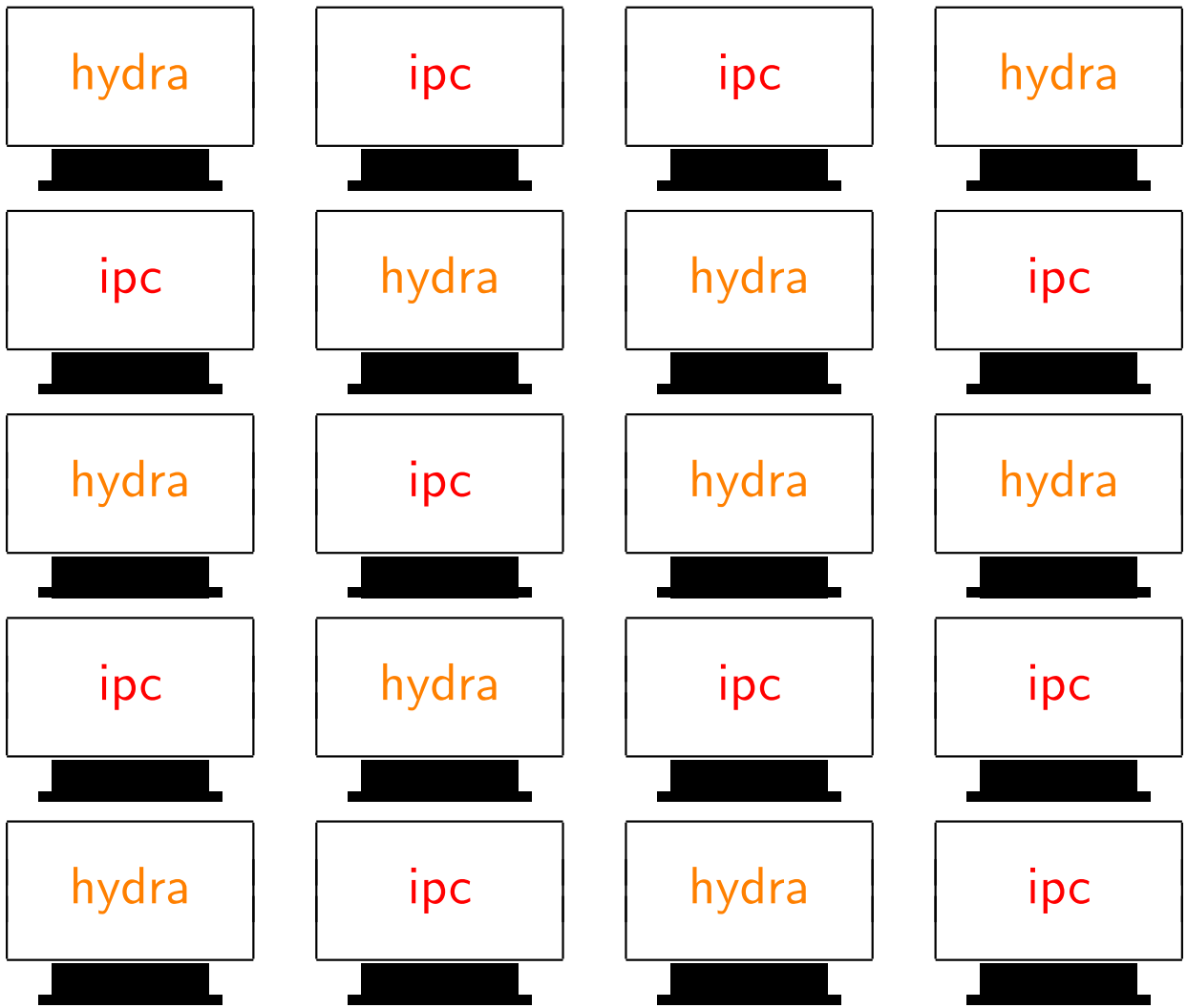
Workflow: ipc, Hydra and Uniform



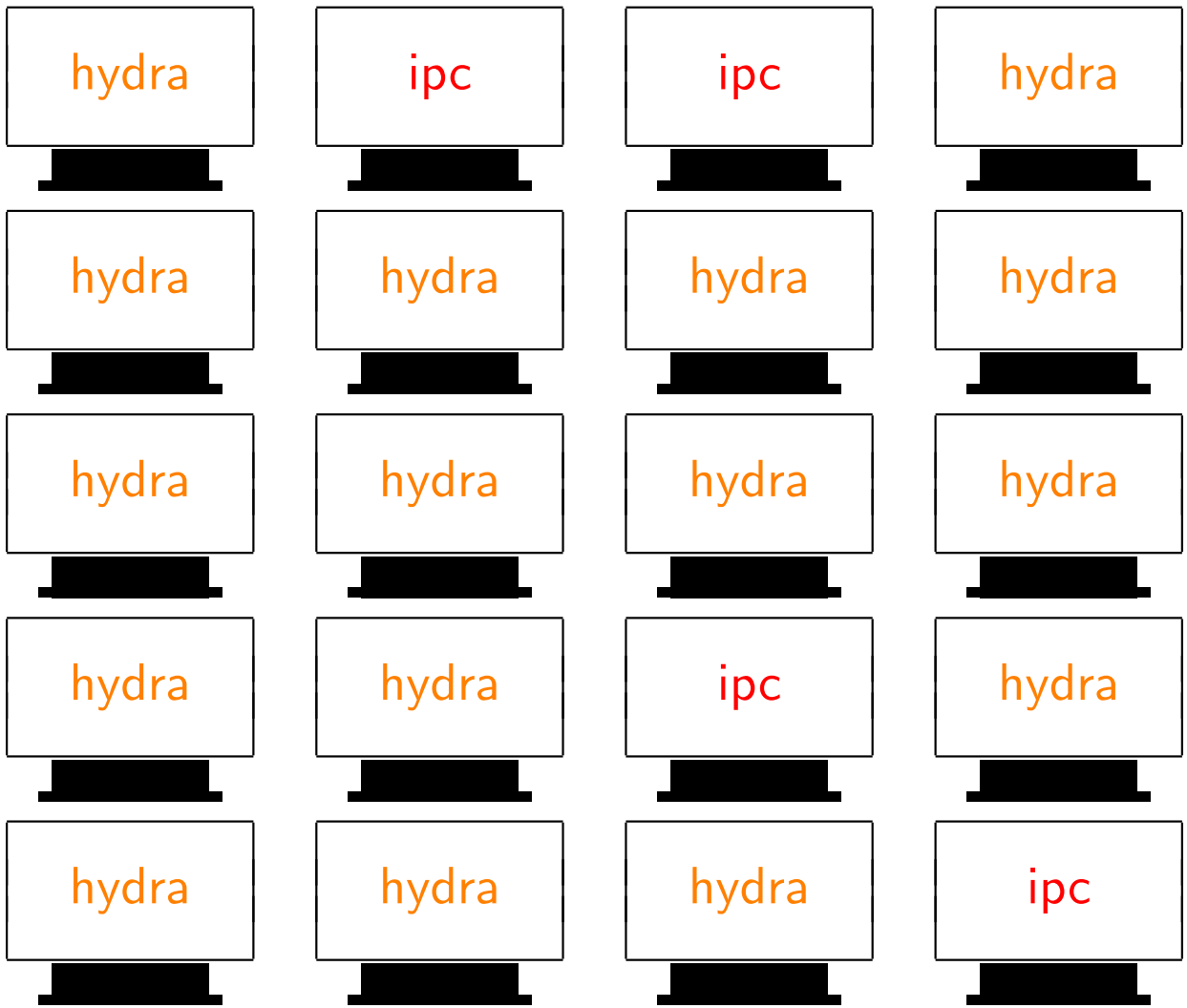
ipc, Hydra and Uniform on Condor



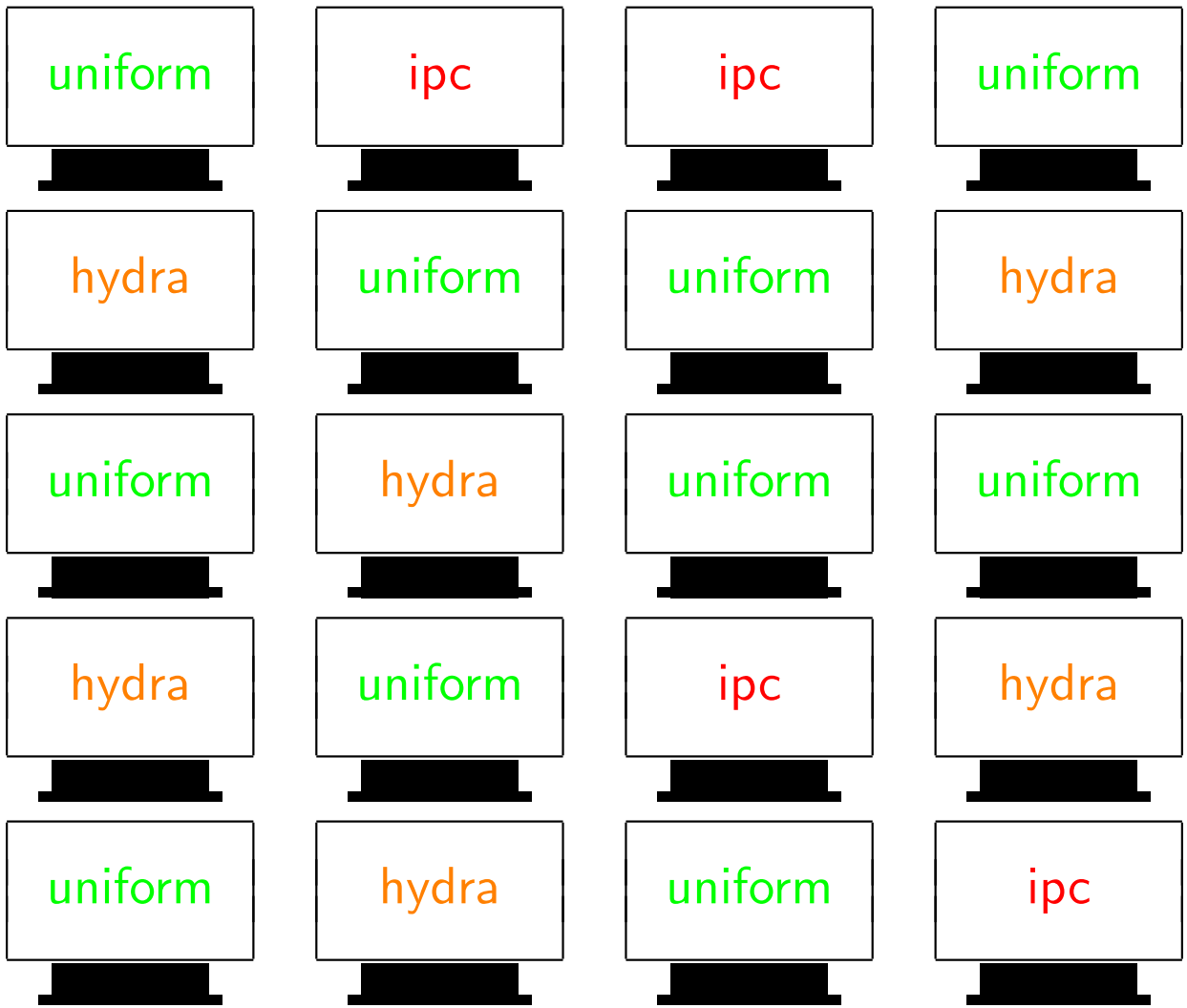
ipc, Hydra and Uniform on Condor



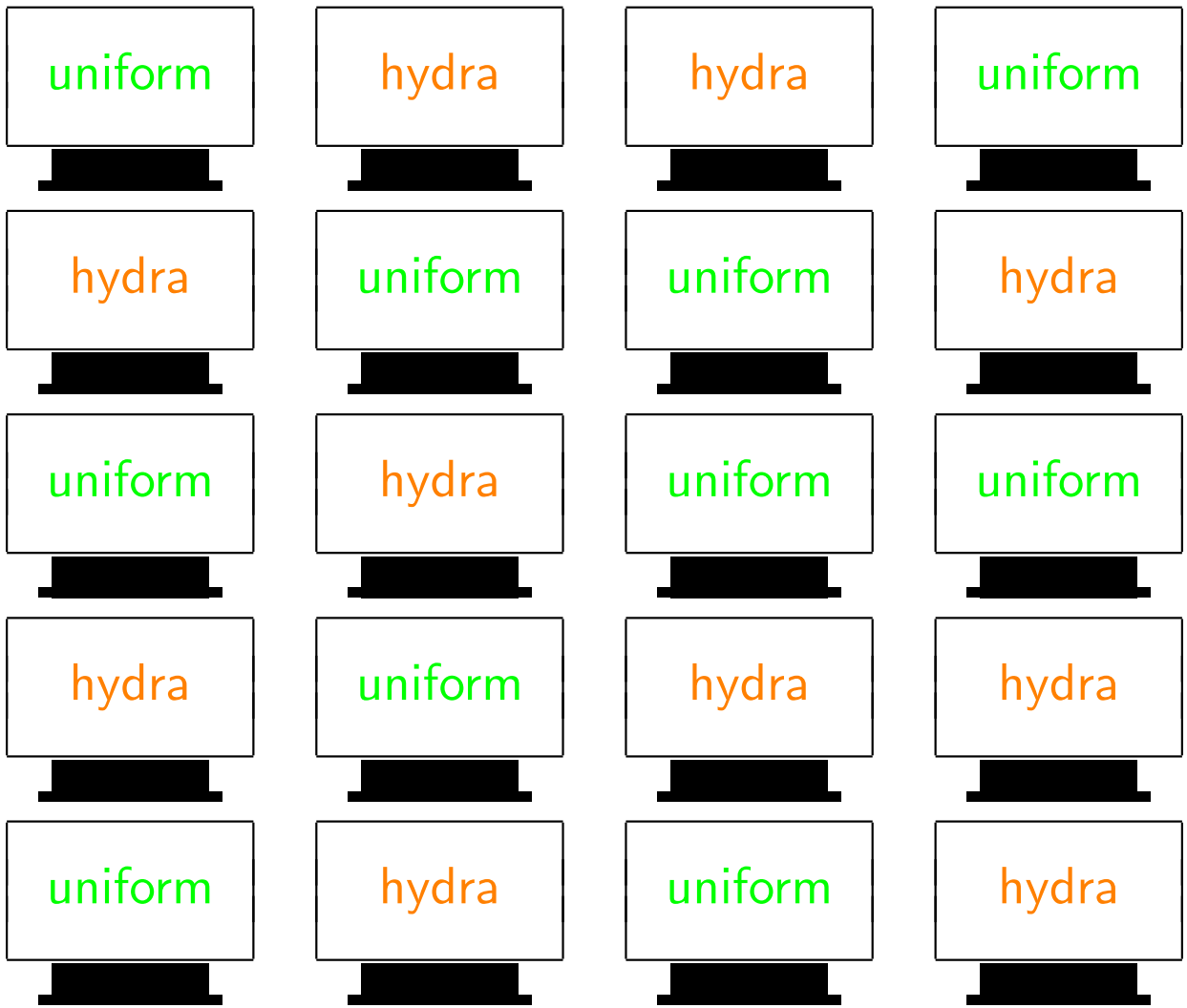
ipc, Hydra and Uniform on Condor



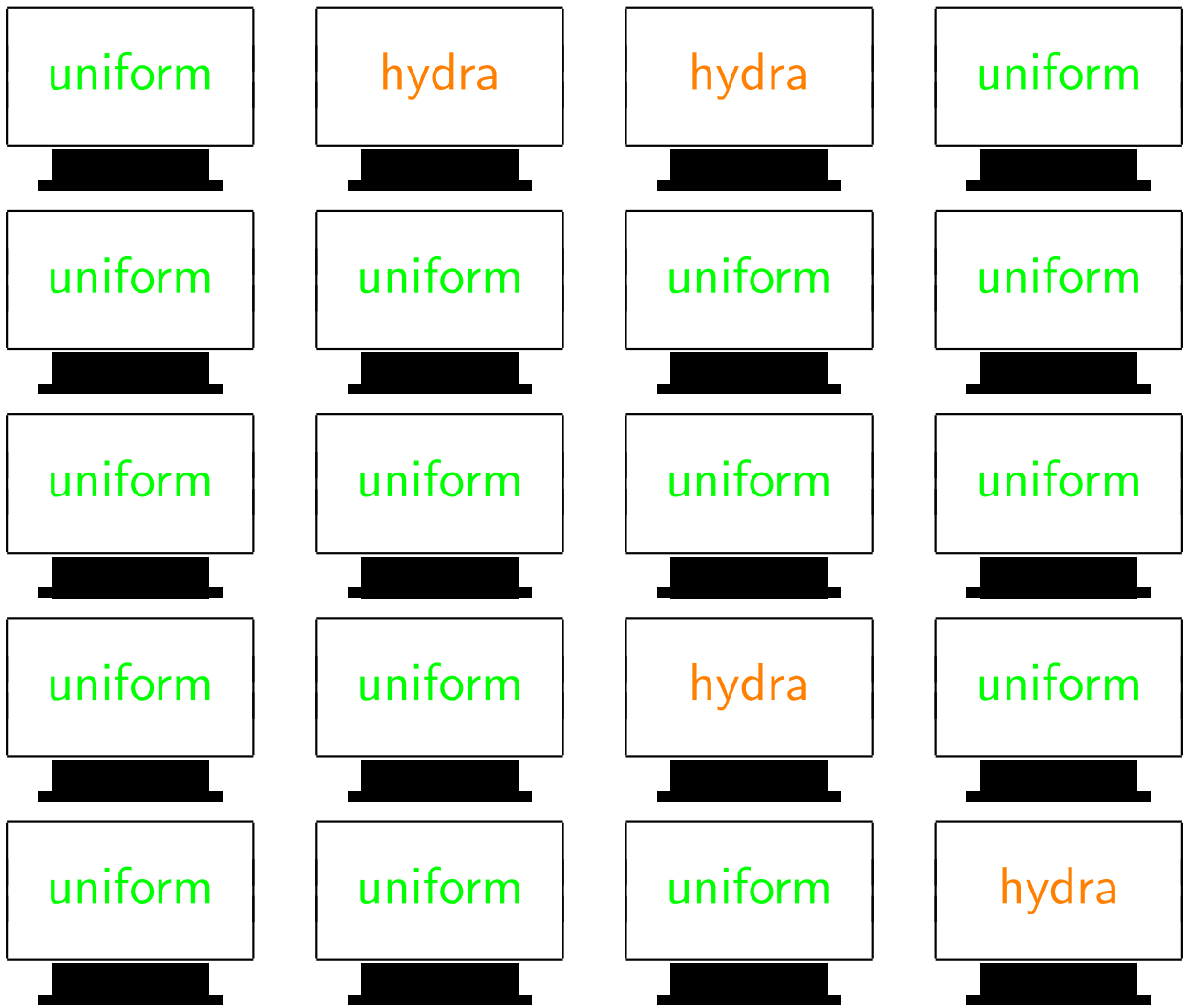
ipc, Hydra and Uniform on Condor



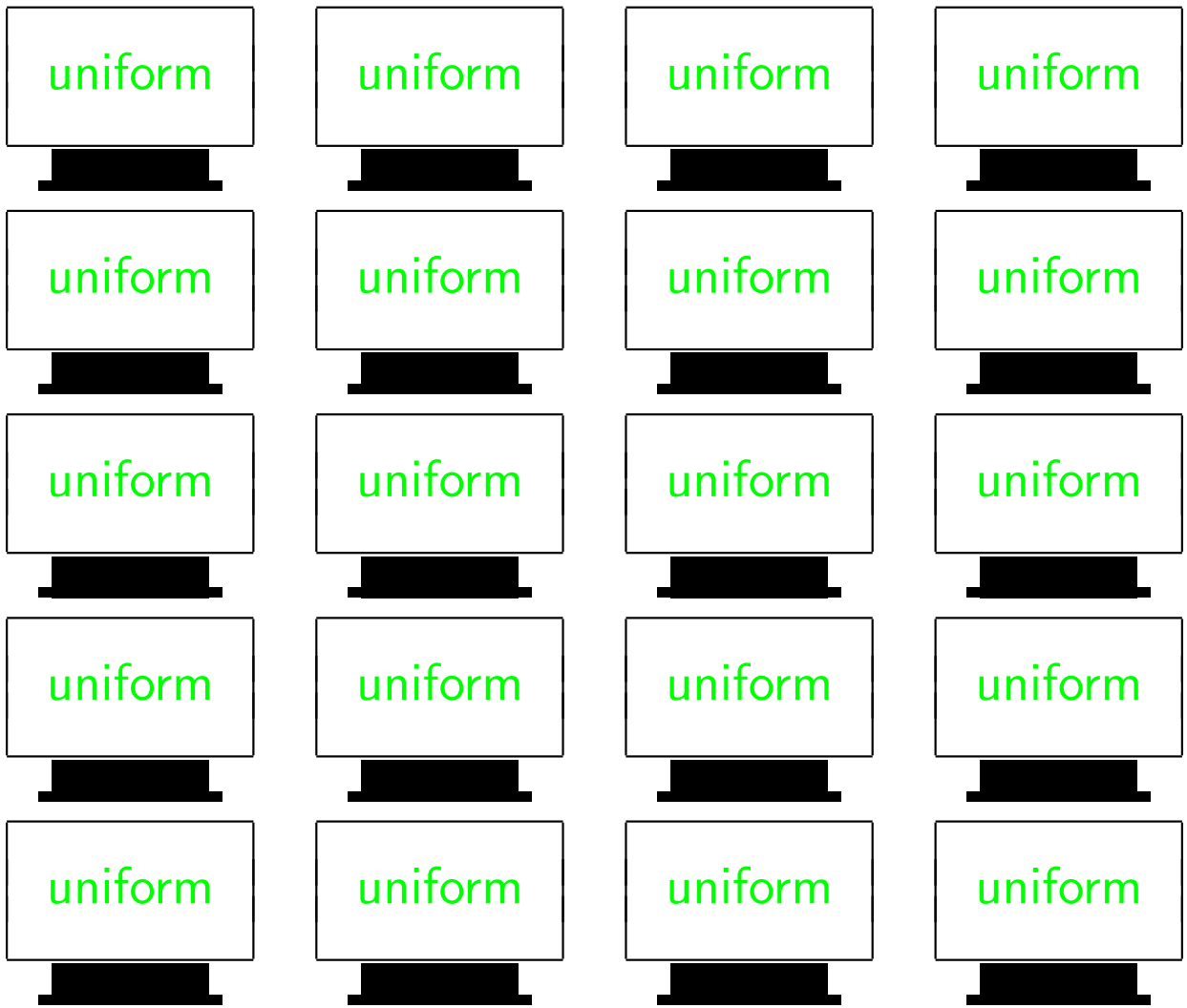
ipc, Hydra and Uniform on Condor



ipc, Hydra and Uniform on Condor



ipc, Hydra and Uniform on Condor

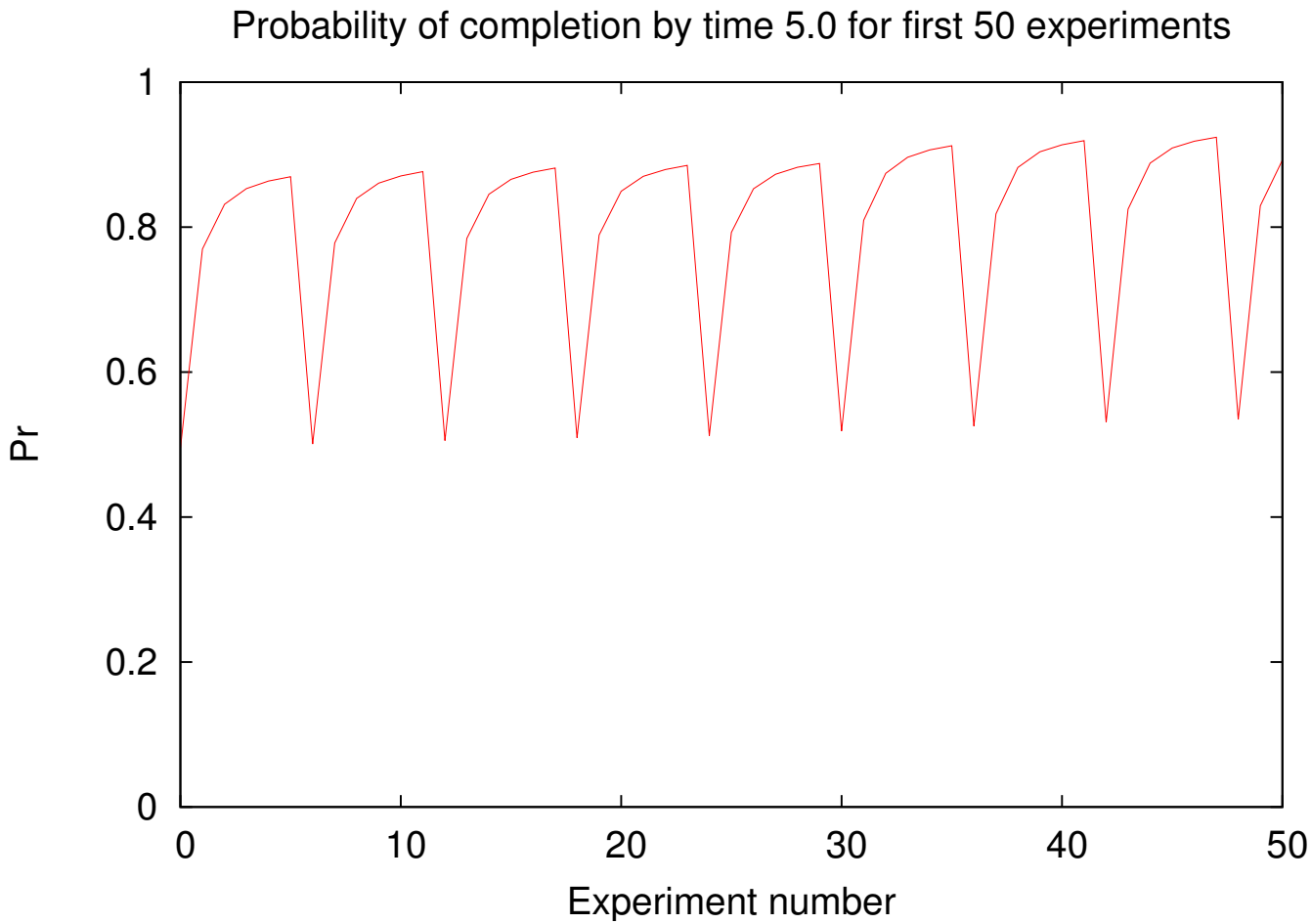


Puffball

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

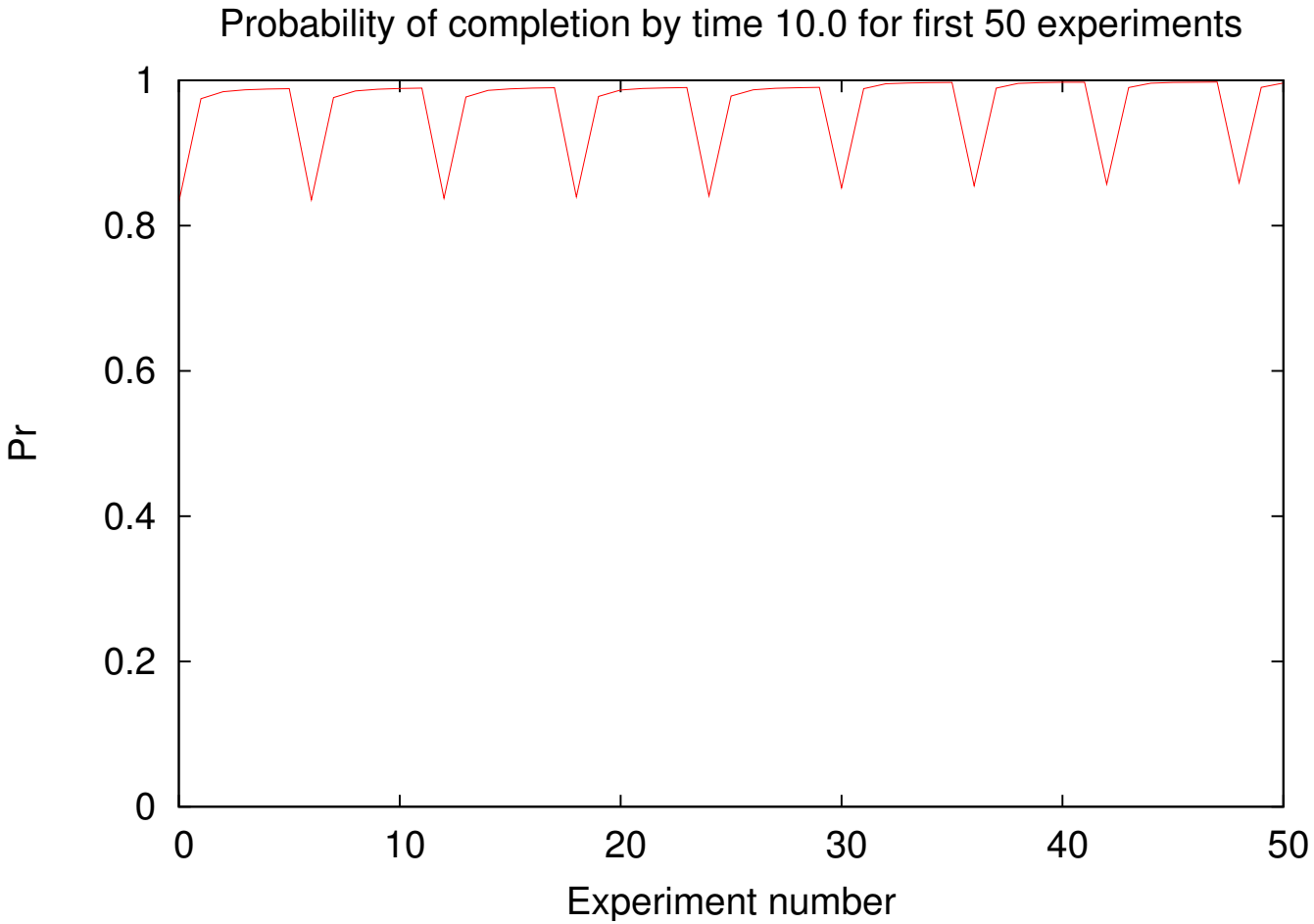
Probability that help is dispatched in 5 or 10 mins

The first 50 experiments (five minute time bound)

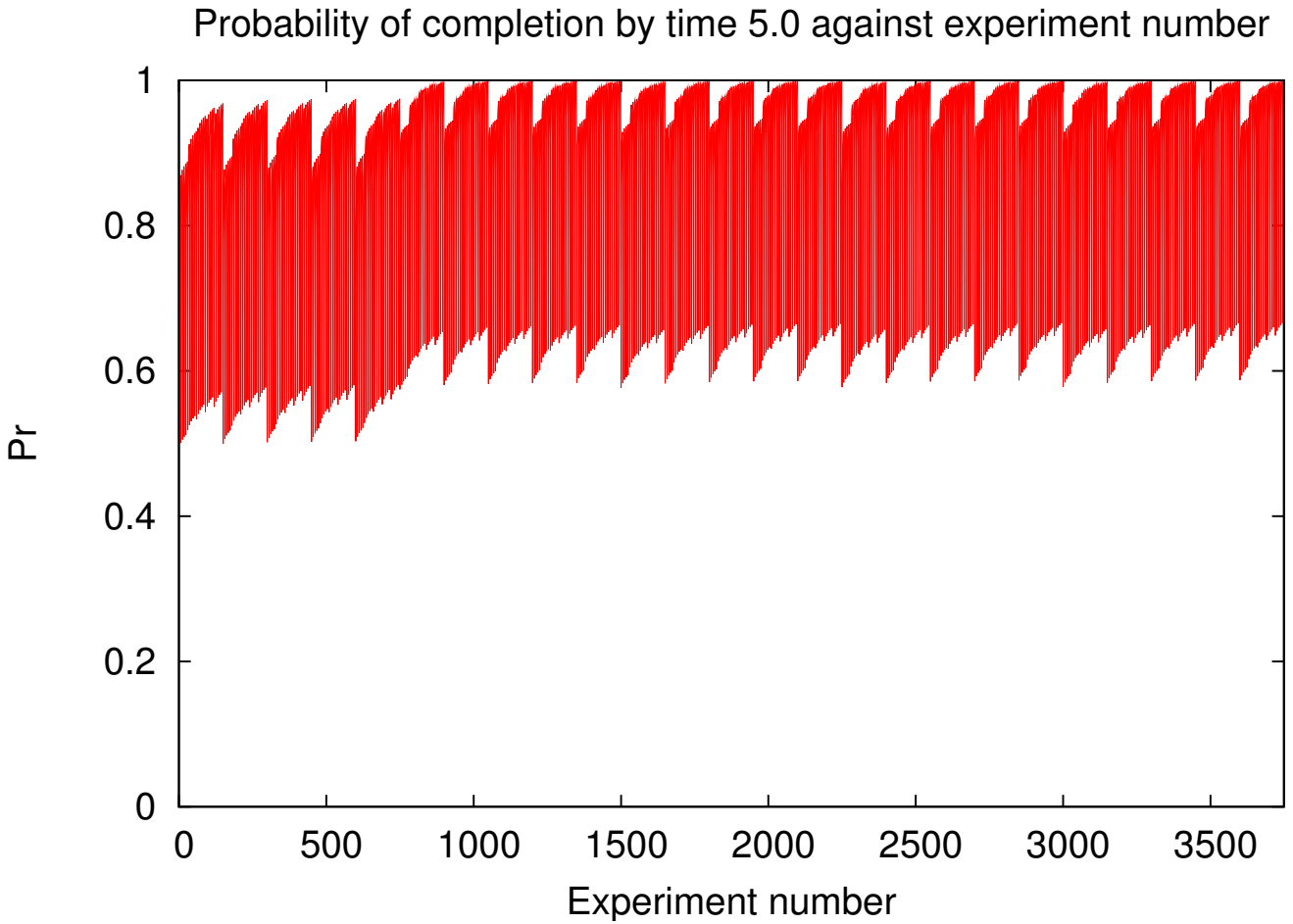


Probability that help is dispatched in 5 or 10 mins

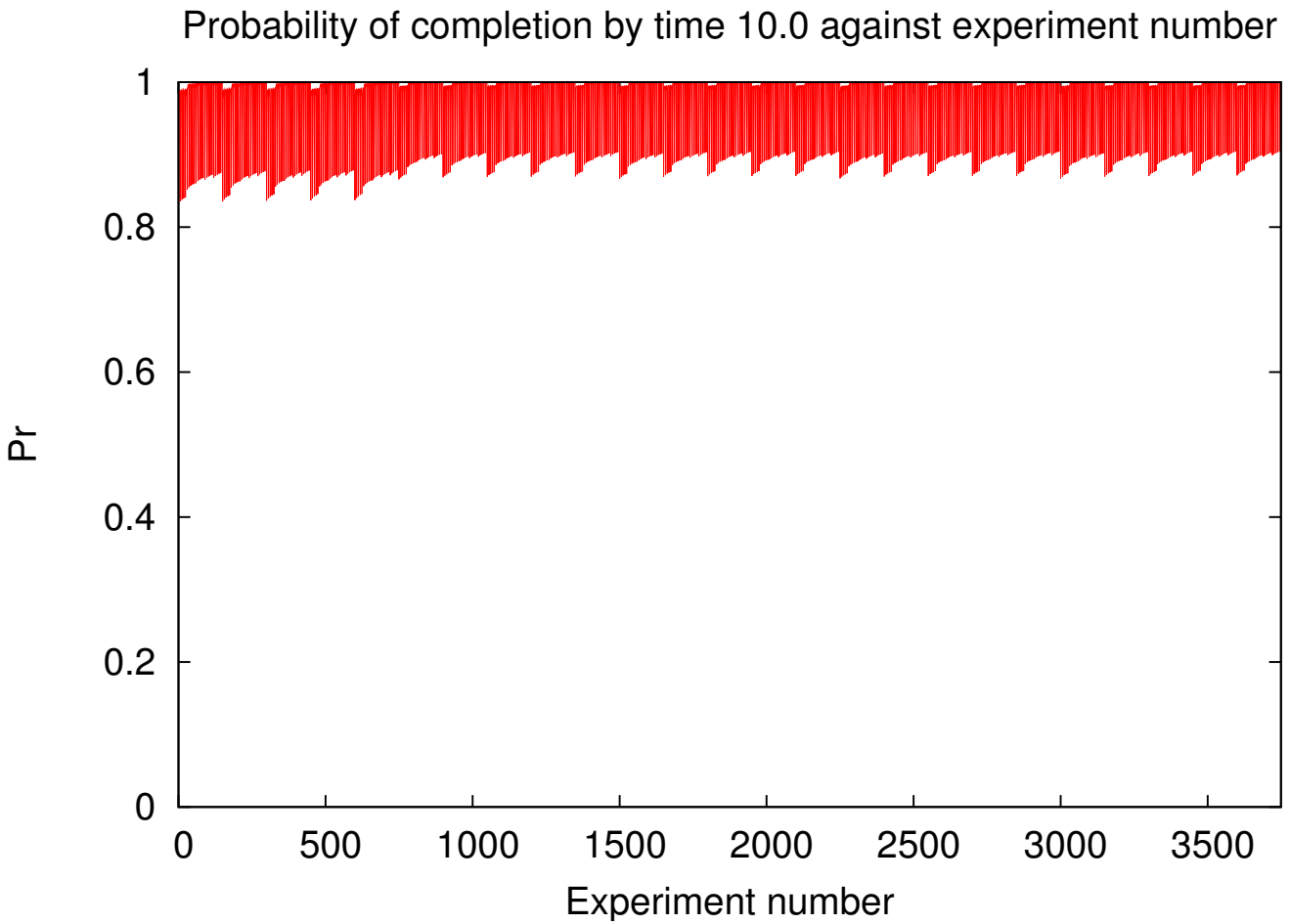
The first 50 experiments (ten minute time bound)



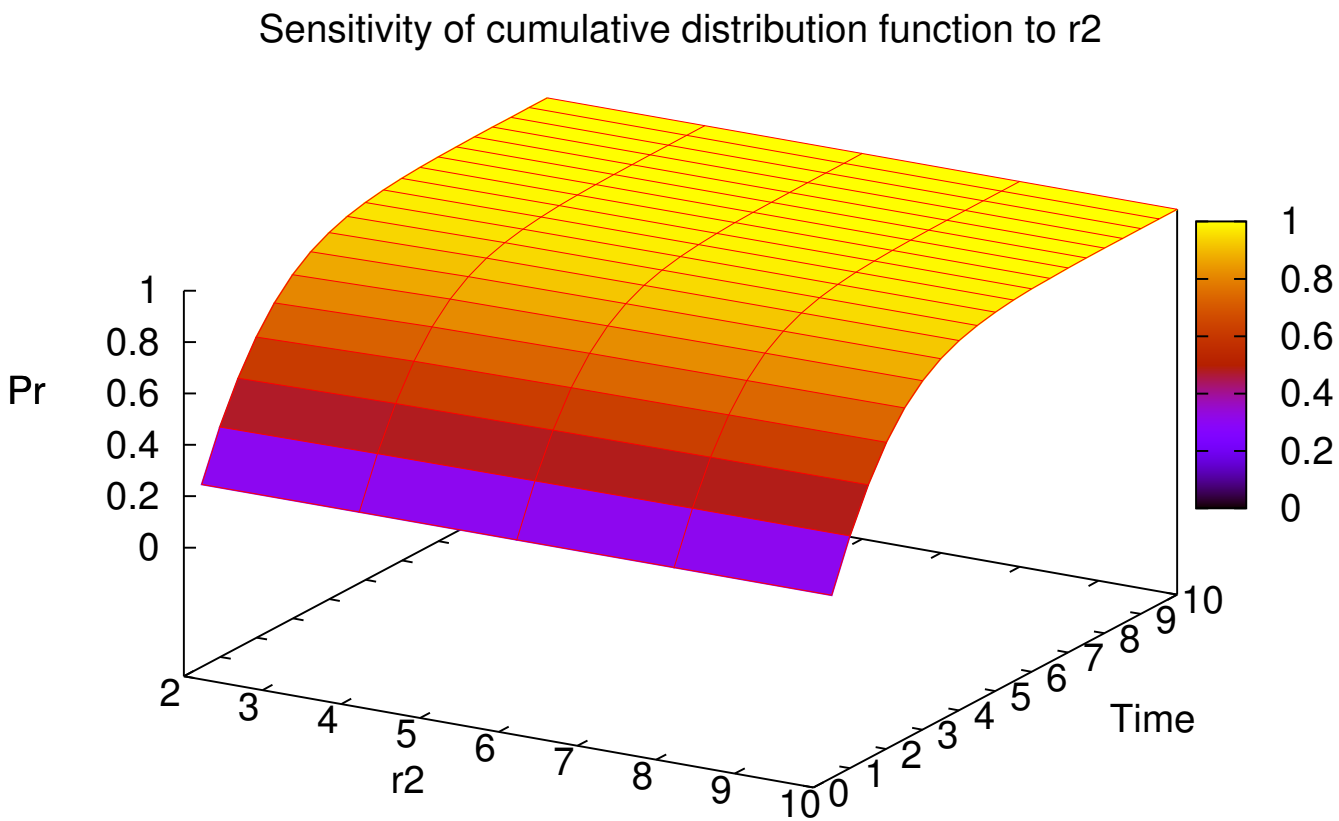
All 3750 experiments (five minute time bound)



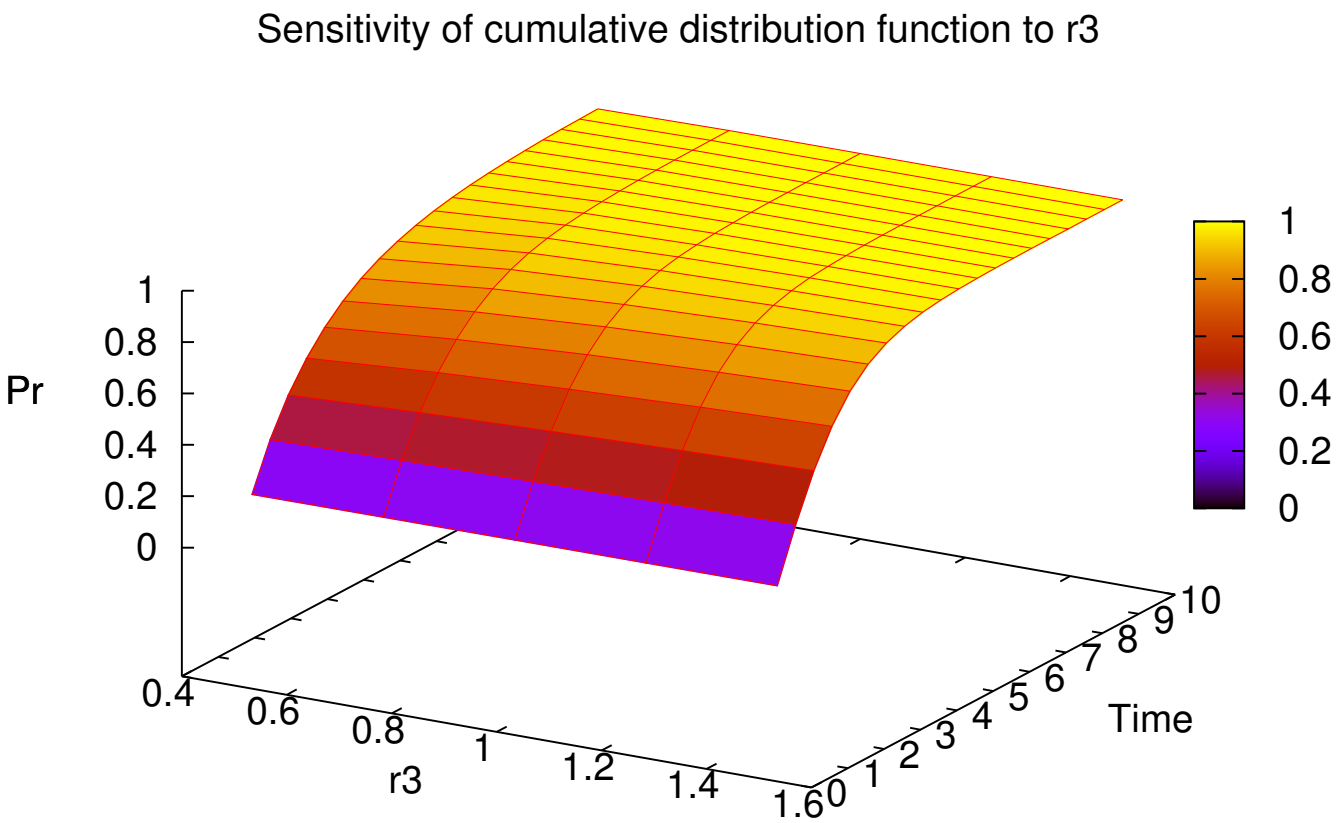
All 3750 experiments (ten minute time bound)



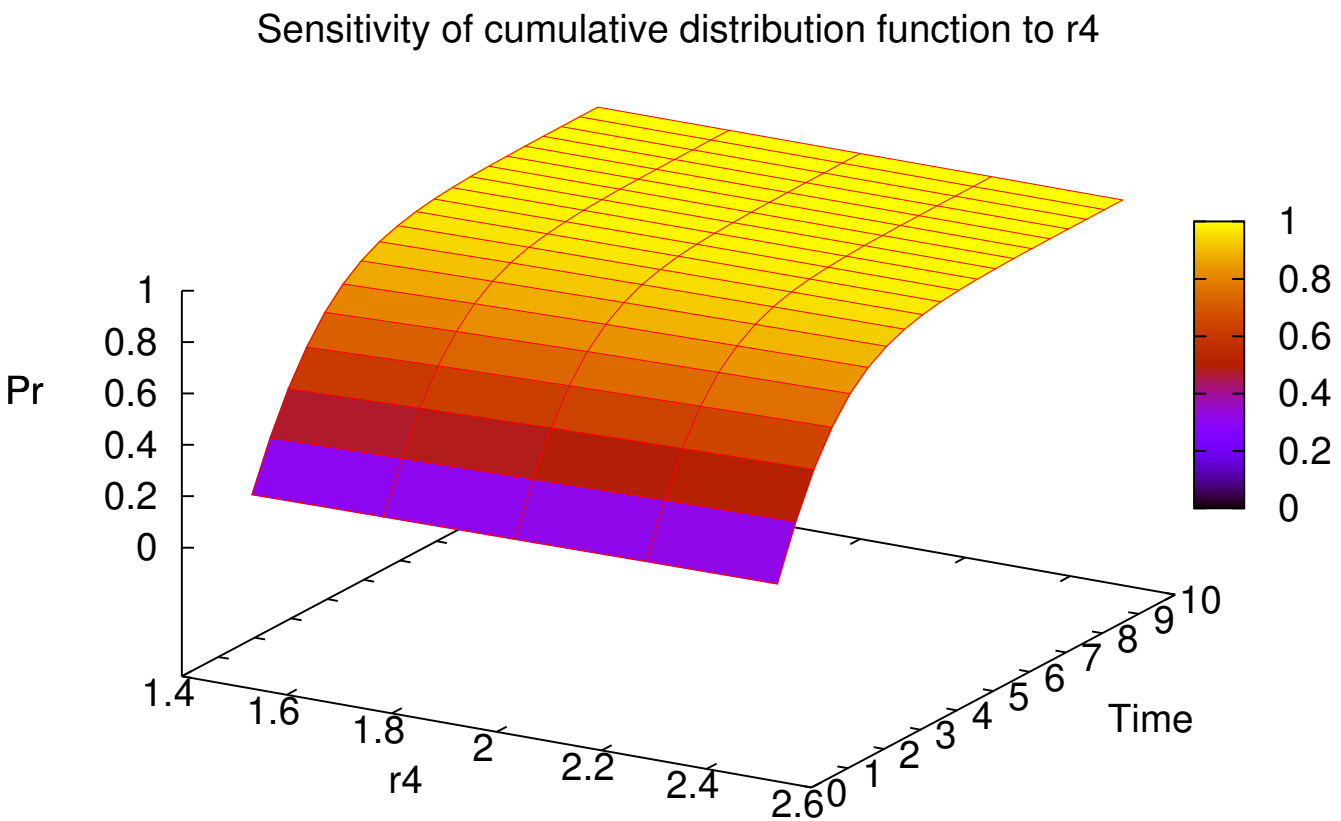
Sensitivity analysis on r_2



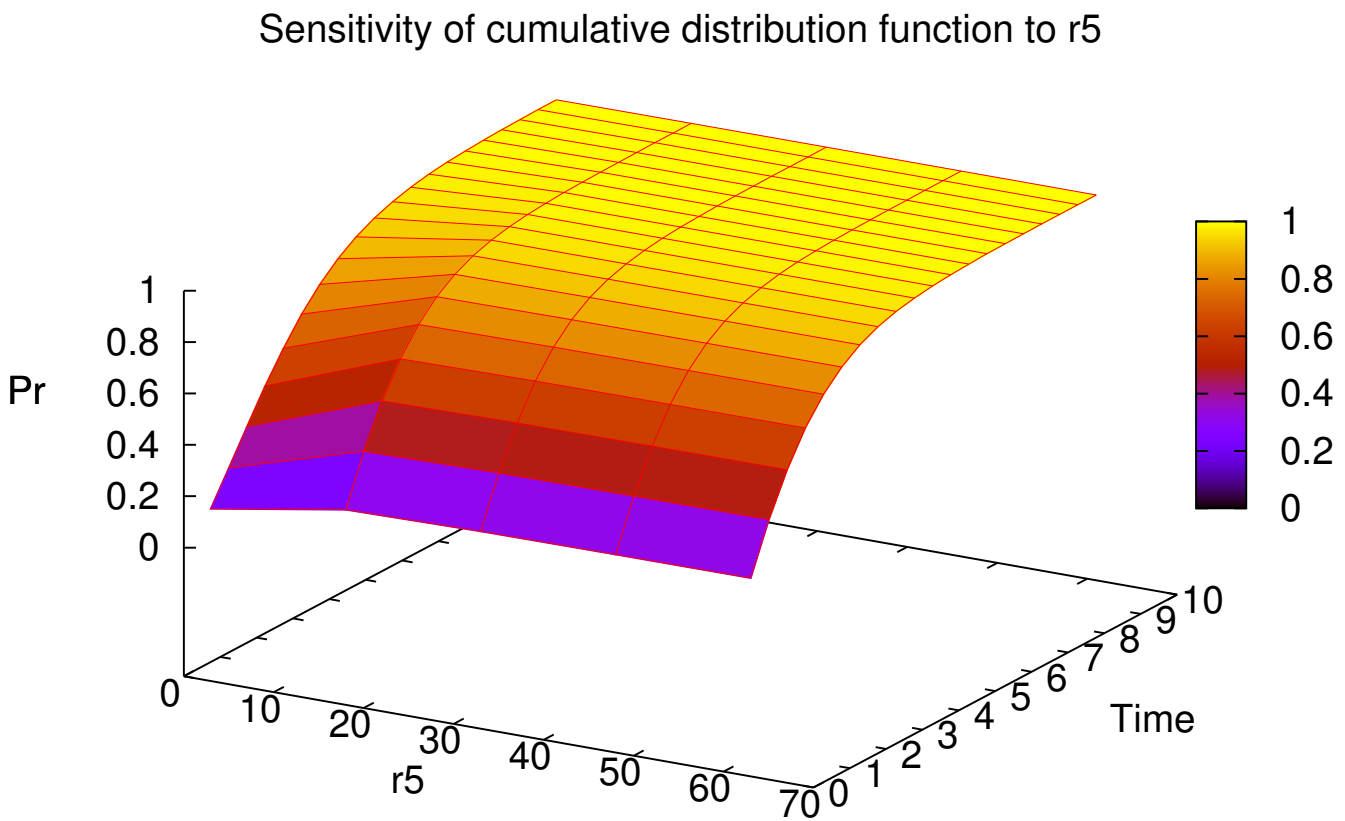
Sensitivity analysis on r_3



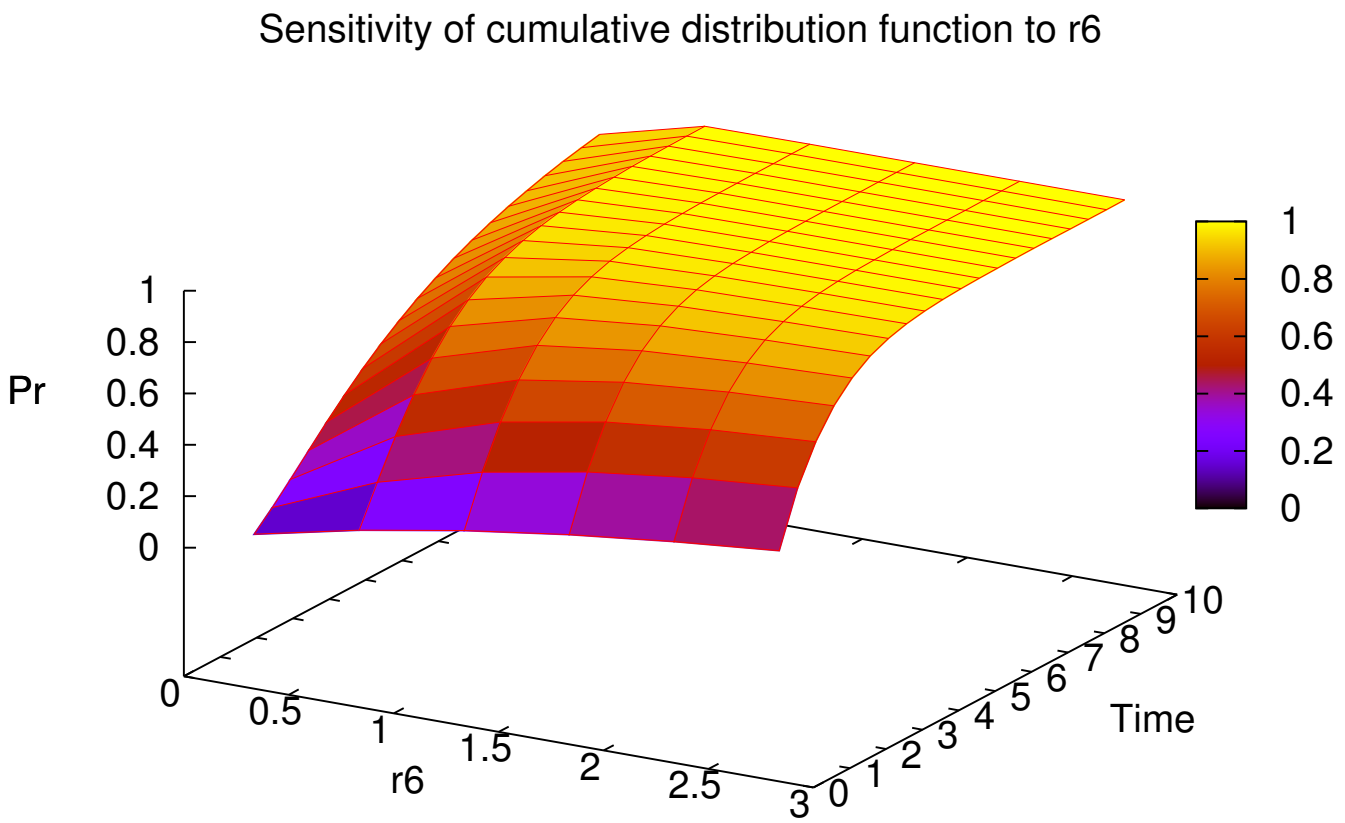
Sensitivity analysis on r_4



Sensitivity analysis on r_5



Sensitivity analysis on r_6



Conclusions

- Dealing with uncertainty in the rate values in our model required us to perform our analysis many times for a wide range of combinations of values.
- We modified our process algebra and Markov chain solvers to run on Condor as well as on a single machine and developed a new tool for post-processing the results.
- Using a distributed computing platform seems to be a promising approach to do many analysis runs of stochastic process algebra models in a lightweight way.

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 and made freely available by Jeremy Bradley of Imperial College, London.
- 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.