Performance Analysis and Resource Optimisation of Critical Systems Modelled by Petri Nets

Ricardo J. Rodríguez

Ph.D. DISSERTATION

Advisors: Dr. **Jorge Emilio Júlvez Bueno** and Dr. **José Javier Merseguer Hernáiz**

Dpto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza

June 24th, 2013

Zaragoza, Spain

Motivation (I): What?

Vulnerable Critical Systems

- Mission must be accomplished in hostile environments
  - **Vulnerable**
  - Internal or external faults
- E.g. SCADA systems, avionics, aerospace, manufacturing, logistics. . .

Motivation (I): What?

Vulnerable Critical Systems

- Mission must be accomplished in hostile environments
  - **Vulnerable**
  - Internal or external faults
- E.g. SCADA systems, avionics, aerospace, manufacturing, logistics. . .

Motivation (I): What?

Vulnerable Critical Systems

- Mission must be accomplished in hostile environments
  - **Vulnerable**
  - Internal or external faults
- E.g. SCADA systems, avionics, aerospace, manufacturing, logistics. . .


Motivation (I): What?

Vulnerable Critical Systems

- Mission must be accomplished in hostile environments
  - **Vulnerable**
  - Internal or external faults
- E.g. SCADA systems, avionics, aerospace, manufacturing, logistics...

Fault-Tolerant Systems

- Critical systems that add FT mechanisms to deal with faults
  - E.g., watchdog, rollback, rollforward, etc.
- Shared resources, naturally modelled as Resource Allocation System (RAS) → resources can be compromised or fail
- Modelled by UML as semi-formal model language
- Petri nets (PNs) as formal model language

Motivation (II): What else?

Expressing Requirements when Designing a System...

- Requirement: a property of the system

Motivation (II): What else?

Expressing Requirements when Designing a System...

- Requirement: a property of the system
  - Functional: how the system is supposed to behave
    - Technical data, data processing...
  - Non-functional (NFP): how the system is supposed to perform
    - Usability, performance, reliability...
    - E.g: how many customers can be attended by a web service? how many bytes can be transferred? how many bytes can be ciphered?

Motivation (III): Where does security fit in?

Security: the Forgotten One

- Non-functional property of the system
- Lack of interest
- Consequence: "fix it later"
  - Fix the problem when the problem arises...

Motivation (III): Where does security fit in?

Security: the Forgotten One

- Non-functional property of the system
- Lack of interest
- Consequence: "fix it later"
  - Fix the problem when the problem arises...
- Severe consequences
  - High cost reimplementation/redesign
  - Economic losses
  - Down services → less customers
  - Disclosure of confidential data (e.g., Sony PSN, RSA company)

Motivation (III): Where does security fit in?

Security: the Forgotten One

- Non-functional property of the system
- Lack of interest
- Consequence: "fix it later"
  - Fix the problem when the problem arises...
- Severe consequences
  - High cost reimplementation/redesign
  - Economic losses
  - Down services → less customers
  - Disclosure of confidential data (e.g., Sony PSN, RSA company)

Security: from the beginning to the end

Motivation (IV): Considering Security in Early Stages

Main Contributions of this Dissertation

SecAM: A UML Profile for Security Analysis and Modelling

- UML tailored for specific purposes: profiling
- Cryptography, Access Control, Security Mechanisms, . . .
- Enables quantitative analysis

Motivation (IV): Considering Security in Early Stages

Main Contributions of this Dissertation

SecAM: A UML Profile for Security Analysis and Modelling

- UML tailored for specific purposes: profiling
- Cryptography, Access Control, Security Mechanisms, ...
- Enables quantitative analysis

Fault-Tolerant Techniques models

- Using UML and Petri nets (PNs)
- Composable models
- Allows to introduce different security models
  - Find appropriate security strategies while meeting performance requirements
- From UML to PNs to analyse

Motivation (V): Just add security, fine. . . anything else?

- Security addition to a system is not free

Motivation (V): Just add security, fine. . . anything else?

- Security addition to a system is not free
- Can affect other NFPs
  - E.g. the more security, the lower system performance
- Other trade-offs exist: availability, usability, economic
- Our main interest: performance
  - Throughput: jobs completed per unit of time

Motivation (V): Just add security, fine. . . anything else?

- Security addition to a system is not free
- Can affect other NFPs
  - E.g. the more security, the lower system performance
- Other trade-offs exist: availability, usability, economic
- Our main interest: performance
  - Throughput: jobs completed per unit of time

Contribution: A Model-based Methodology

- Quantitative estimation of performance-security trade-off
- Fault-Tolerant Techniques (FTTs) + Security Mechanisms (SMs)
  - FTTs: watchdog, switch over failing, proactive-reactive recovery, . . .
  - SMs: (de)encryption, digital signature generation and verification, . . .
- Goal: To compose security models with software architectural models

Motivation (VI): The BUT. . . (1)

- Recall: FT systems modelled with Petri nets (PNs)
  - More precisely, process Petri nets (PPNs)
    - Good modelling capabilities
    - Interesting analysis skills. . .

Motivation (VI): The BUT. . . (1)

- Recall: FT systems modelled with Petri nets (PNs)
  - More precisely, process Petri nets (PPNs)
    - Good modelling capabilities
    - Interesting analysis skills. . .
- State explosion problem: sooooo large
  - Computation of performance becomes unachievable

Motivation (VI): The BUT. . . (1)

- Recall: FT systems modelled with Petri nets (PNs)
  - More precisely, process Petri nets (PPNs)
    - Good modelling capabilities
    - Interesting analysis skills. . .
- State explosion problem: sooooo large
  - Computation of performance becomes unachievable
- Avoid exact computation: upper throughput bounds
  - Using Linear Programming (LP) techniques
  - Good accuracy – computational complexity trade-off

Motivation (VI): The BUT. . . (2)

Contribution: Strategies for Upper Thr. Bounds Computation

- Iterative strategies: The *regrowing* approach
- Close to the real throughput
- For Stochastic Marked Graphs and Stochastic Process Petri nets
- Implemented in a tool: `PeabraiN`

Motivation (VI): The BUT. . . (2)

Contribution: Strategies for Upper Thr. Bounds Computation

- Iterative strategies: The *regrowing* approach
- Close to the real throughput
- For Stochastic Marked Graphs and Stochastic Process Petri nets
- Implemented in a tool: `PeabraiN`

Contribution: Data-throttling in Scientific Workflows

- Minimise use of input buffers at tasks
- Better use of network bandwidth
- Also a quantitative metric for evaluating optimisation
- Implemented in a tool: `DT4SW` (not yet released)

Motivation (VI): The BUT. . . (3)

- Recall: security addition is not free
  - Performance, availability, etc. can be influenced

Motivation (VI): The BUT. . . (3)

- Recall: security addition is not free
  - Performance, availability, etc. can be influenced
- FT systems with shared resources
  - Can we add more resources to compensate throughput degradation?

Motivation (VI): The BUT. . . (3)

- Recall: security addition is not free
  - Performance, availability, etc. can be influenced
- FT systems with shared resources
  - Can we add more resources to compensate throughput degradation?
- It depends → resource addition IS NOT free either
  - But let us try something. . .

Motivation (VI): The BUT. . . (3)

- Recall: security addition is not free
  - Performance, availability, etc. can be influenced
- FT systems with shared resources
  - Can we add more resources to compensate throughput degradation?
- It depends → resource addition IS NOT free either
  - But let us try something. . .

Contribution: Compensation of Throughput Degradation

- Several algorithms to compensate throughput degradation
- Subject to a (major) constraint: Budget

Motivation (VI): The BUT. . . (3)

- Recall: security addition is not free
  - Performance, availability, etc. can be influenced
- FT systems with shared resources
  - Can we add more resources to compensate throughput degradation?
- It depends → resource addition IS NOT free either
  - But let us try something. . .

Contribution: Compensation of Throughput Degradation

- Several algorithms to compensate throughput degradation
- Subject to a (major) constraint: Budget
  - A LP problem: Maximise throughput via incrementing resources
  - An Integer-LP problem: Minimise cost of compensating, considering either:
    - Number of resources
    - Timing of FT techniques added

Security Modelling (I): UML profiling

UML profile: what?

- OMG standard
- Stereotypes and tagged values to annotate UML elements

Security Modelling (I): UML profiling

UML profile: what?

- OMG standard
- Stereotypes and tagged values to annotate UML elements
  - Expressing Non-Functional Properties (NFP) on the UML designs
  - Extending model semantics

Security Modelling (I): UML profiling

UML profile: what?

- OMG standard
- Stereotypes and tagged values to annotate UML elements
  - Expressing Non-Functional Properties (NFP) on the UML designs
  - Extending model semantics

Some examples

- *Modelling and Analysis of RT Embedded systems* (MARTE)
  - Support for performance and schedulability analysis
  - NFPs expressed thru VSL (*Value Specification Language*) syntax

Security Modelling (I): UML profiling

UML profile: what?

- OMG standard
- Stereotypes and tagged values to annotate UML elements
  - Expressing Non-Functional Properties (NFP) on the UML designs
  - Extending model semantics

Some examples

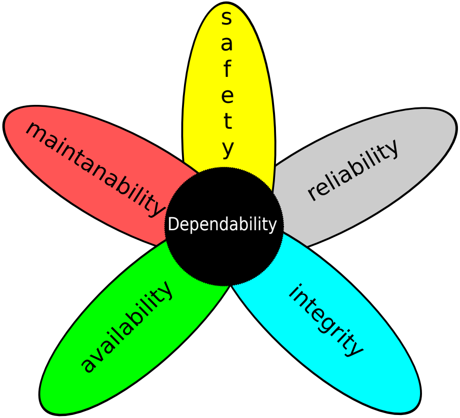
- *Modelling and Analysis of RT Embedded systems* (MARTE)
  - Support for performance and schedulability analysis
  - NFPs expressed thru VSL (*Value Specification Language*) syntax
- Dependability Analysis and Modelling (DAM)
  - MARTE specialisation
  - Dependability properties into UML

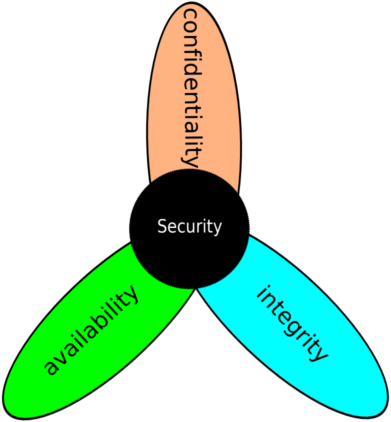
OMG. A UML profile for Modeling and Analysis of Real Time Embedded Systems (MARTE). *Document ptc/09-11-02*, 2009

Bernardi, S. et al. Model-driven Dependability Assessment of Software Systems. *Springer*, 2013

Security Modelling (II): What is security indeed?

Tight relation with *dependability* (Avizienis)

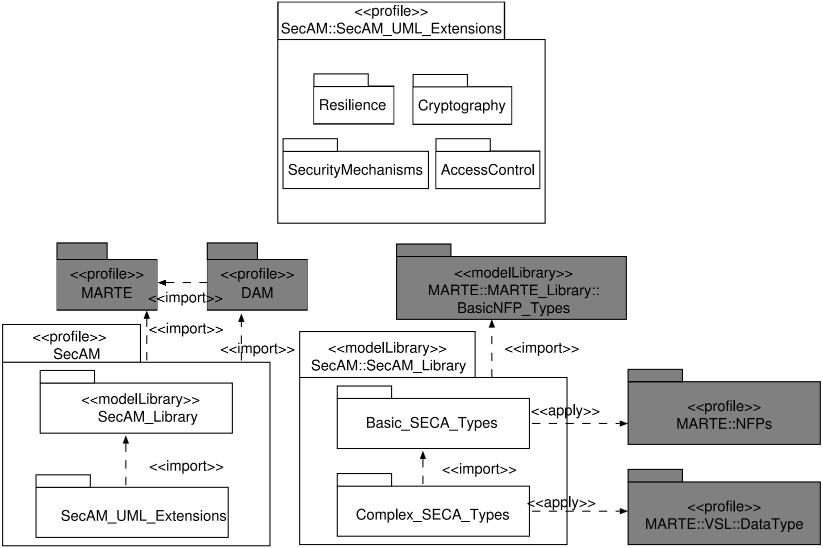




Avizienis, A. et al. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dep. Sec. Comp.*, 2004

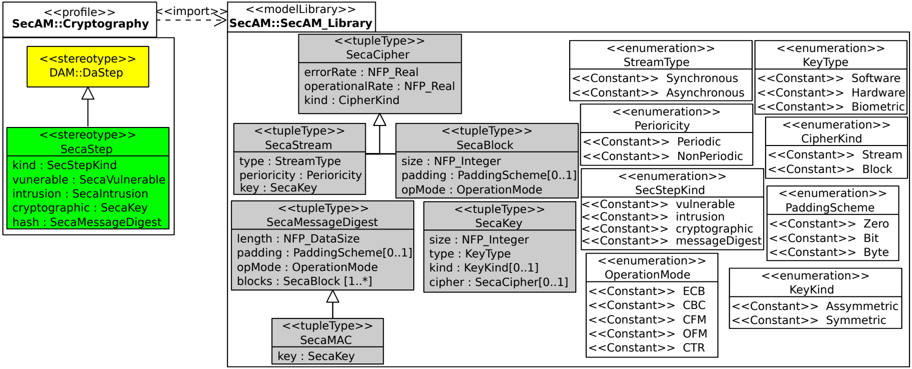
Security Modelling (III): The SecAM profile (1)

Security Analysis and Modelling: A general overview. . .



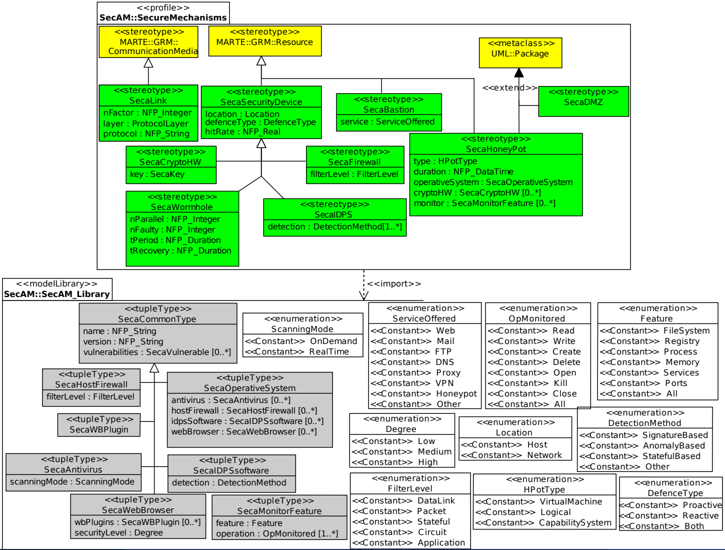
Security Modelling (III): The SecAM profile (2)

Cryptography package



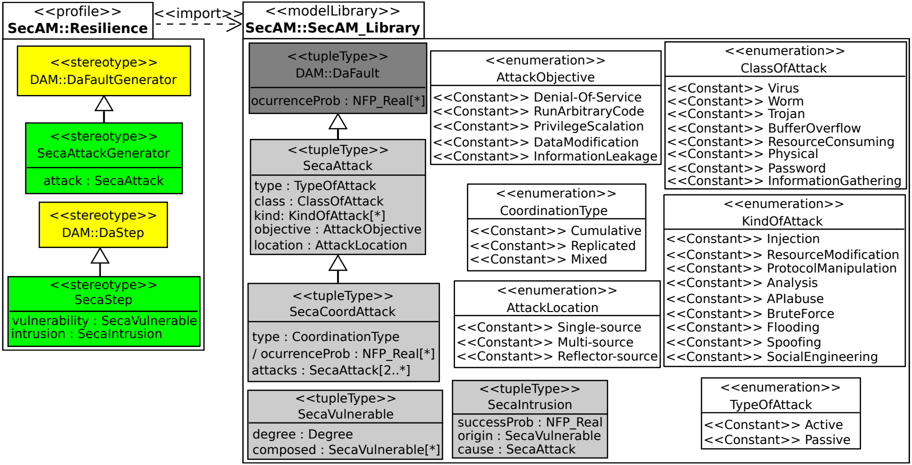
Security Modelling (III): The SecAM profile (3)

SecurityMechanisms package



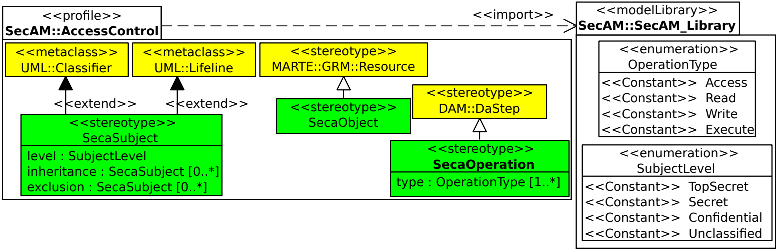
Security Modelling (III): The SecAM profile (3)

Resilience package



Security Modelling (III): The SecAM profile (4)

AccessControl package

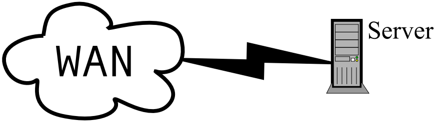


Security Modelling (IV): A Case Study (1)

Problem description

Problem

- Services on-demand system
- 2 kind of services
  - *Service 1*: 1s
  - *Service 2*: 2s
- Maximum of simultaneous requests: 100

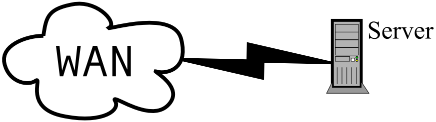


Security Modelling (IV): A Case Study (1)

Problem description

Problem

- Services on-demand system
- 2 kind of services
    - *Service 1*: 1s
    - *Service 2*: 2s
- Maximum of simultaneous requests: 100
- Legitimate and illegitimate users

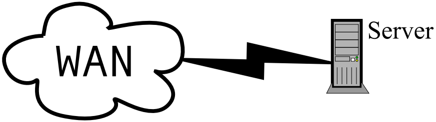


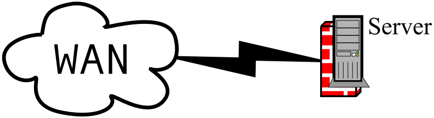
Security Modelling (IV): A Case Study (1)

Problem description

Problem

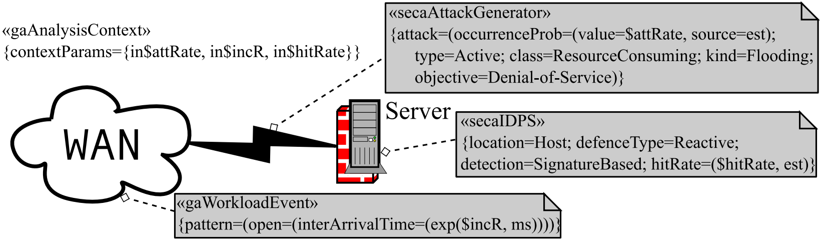
- Services on-demand system
- 2 kind of services
    - *Service 1*: 1s
    - *Service 2*: 2s
- Maximum of simultaneous requests: 100
- Legitimate and illegitimate users





Security Modelling (IV): A Case Study (2)

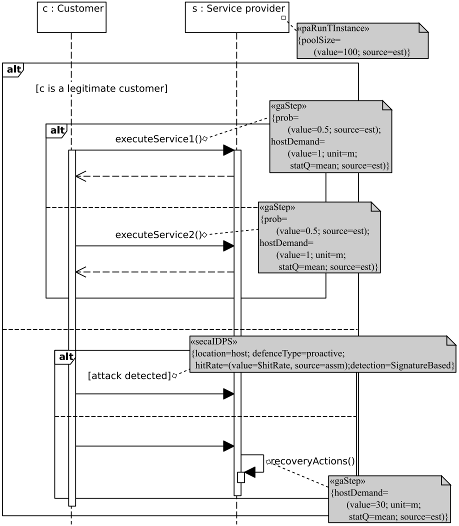
Using SecAM: Season to taste the UML model



- 2 possibilities:
    - IDPS1 (hit rate 80%)
    - IDPS2 (hit rate 95%)

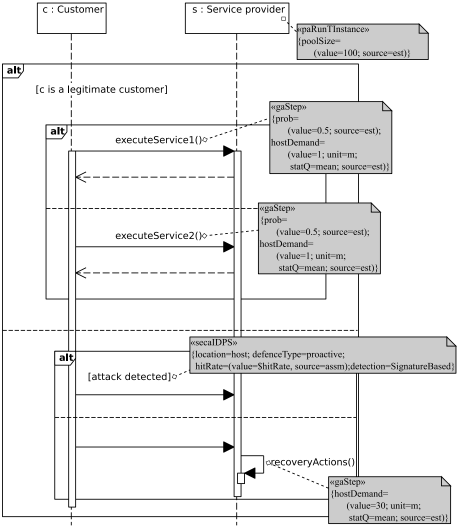
Security Modelling (IV): A Case Study (3)

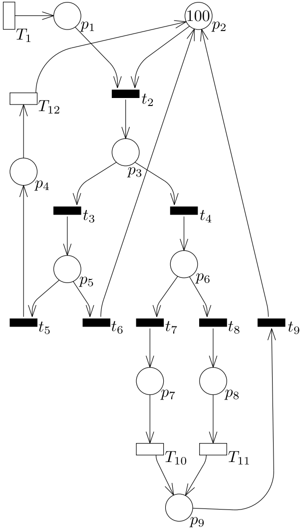
More models. . .



Security Modelling (IV): A Case Study (3)

More models. . .





Security Modelling (IV): A Case Study (4)

Experiments and results

Experiments parameters

- Input customers ratio: $\{5, 10, 20\}$ customers/s
- IDPS hit rate: 80%, 95%
- Attacks rate: $[0.15\% \ldots 37.5\%]$

Security Modelling (IV): A Case Study (4)

Experiments and results

Experiments parameters

- Input customers ratio: $\{5, 10, 20\}$ customers/s
- IDPS hit rate: 80%, 95%
- Attacks rate: $[0.15\% \ldots 37.5\%]$

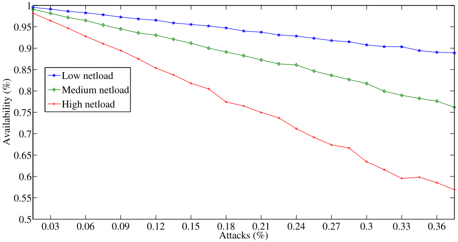


Figure: Hit rate 80%

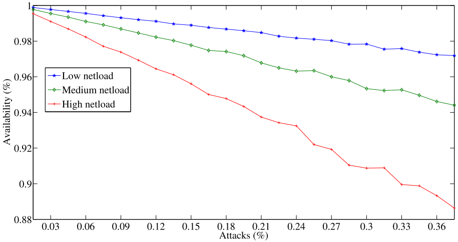


Figure: Hit rate 95%

FTTs Modelling (I): Some Background. . .

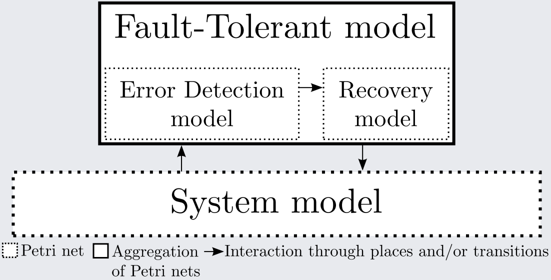
- **Fault Tolerance Techniques**
  - Error Detection
    - *Concurrent Detection*
    - *Preemptive Detection*
  - Recovery
    - Error Handling
      - *Rollback*
      - *Rollforward*
      - *Compensation*
    - Fault Handling
      - *Diagnosis*
      - *Isolation*
      - *Reconfiguration*
      - *Reinitialisation*

- Phases in a FT actuation
- Aims at fault avoidance carrying out error detection and system recovery

Avizienis, A. et al. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dep. Sec. Comp.*, 2004

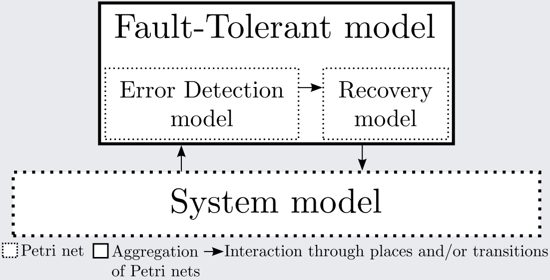
FTTs Modelling (II): Based on PNs (1)

Integration with a system



FTTs Modelling (II): Based on PNs (1)

Integration with a system

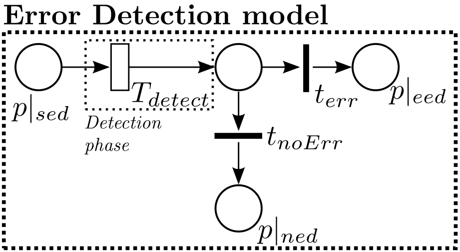


Transformation Rule

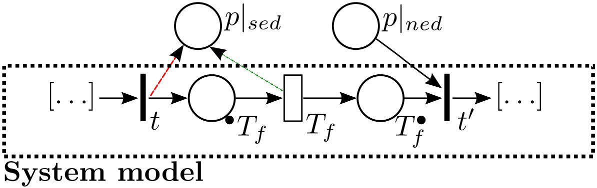
- Transition that fails: T_f
- Adding pre and post places $\rightarrow$ allows model composition

FTTs Modelling (II): Based on PNs (2)

Error Detection model



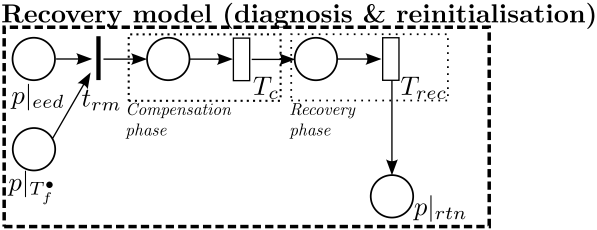
(a) Error Detection model



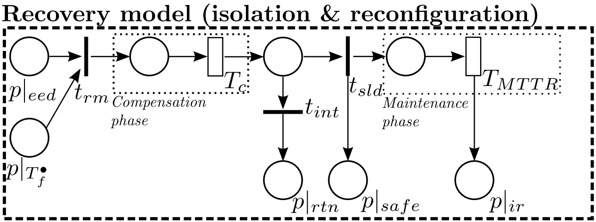
(b) Places $p|_{sed}$, $p|_{ned}$ added to the system model

FTTs Modelling (II): Based on PNs (3)

Recovery model



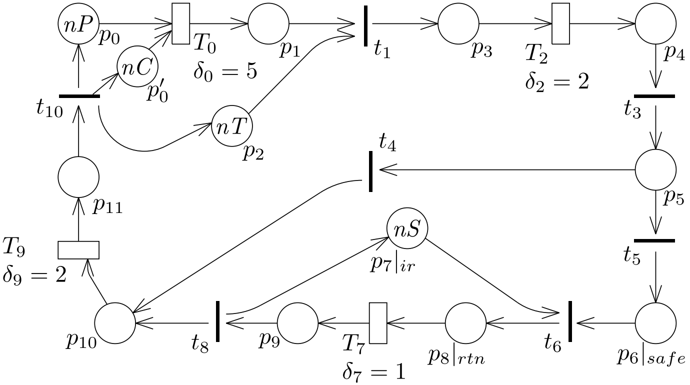
(a) Diagnosis & reinitialisation



(b) Isolation & reconfiguration

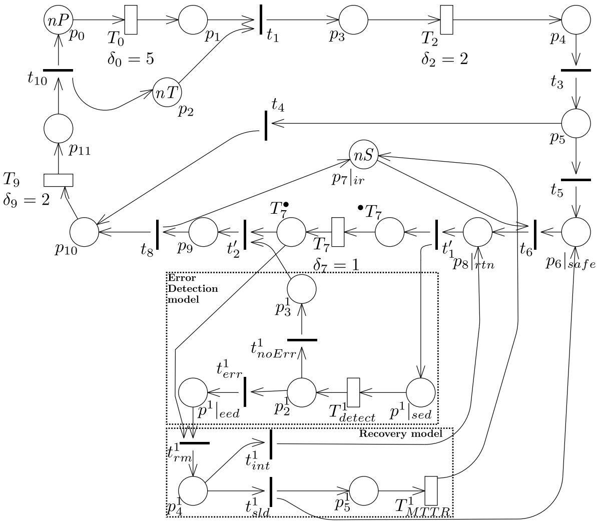
FTTs Modelling (II): Based on PNs (4)

Running Example (1): a packet-routing algorithm



FTTs Modelling (II): Based on PNs (4)

Running Example (2): a packet-routing algorithm with FT capabilities – reconfiguration with rollbackward



FTTs Modelling (III): Based on UML (1)

Fault-Tolerant Techniques (FTTs) model library

- Contains a set of FTTs modelling through UML
- Annotated with MARTE-DAM-`SecAM` profiles

FTTs Modelling (III): Based on UML (1)

Fault-Tolerant Techniques (FTTs) model library

- Contains a set of FTTs modelling through UML
- Annotated with MARTE-DAM-`SecAM` profiles
- Composable models
- Allows to introduce different security models
  - Find appropriate security strategies while meeting performance requirements

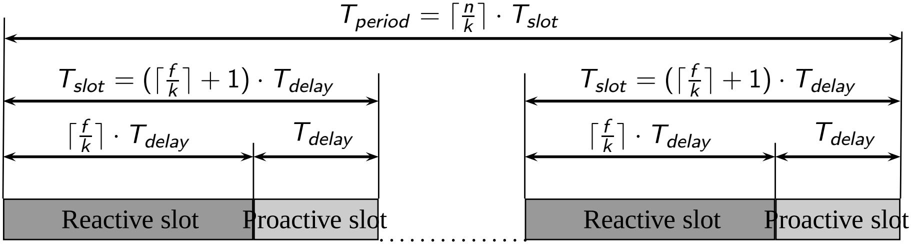
FTTs Modelling (III): Based on UML (1)

Fault-Tolerant Techniques (FTTs) model library

- Contains a set of FTTs modelling through UML
- Annotated with MARTE-DAM-`SecAM` profiles
- Composable models
- Allows to introduce different security models
  - Find appropriate security strategies while meeting performance requirements
- Converted to PNs and analysed

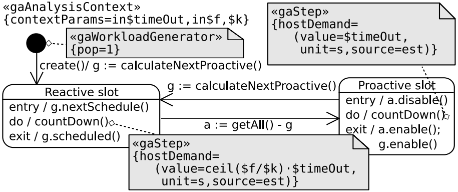
FTTs Modelling (III): Based on UML (2)

A Proactive-Reactive Recovery Technique (1)



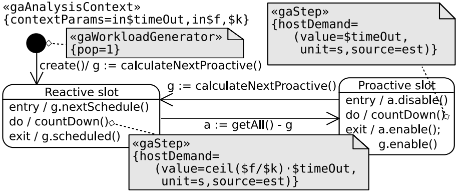
FTTs Modelling (III): Based on UML (2)

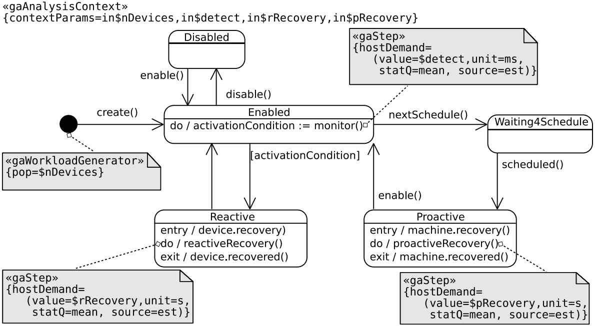
A Proactive-Reactive Recovery Technique (2)



FTTs Modelling (III): Based on UML (2)

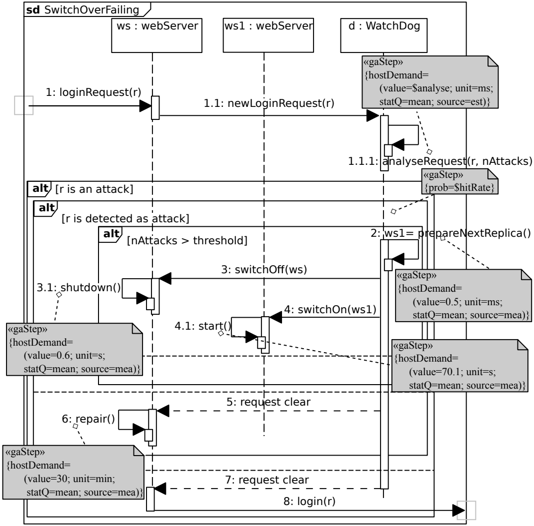
A Proactive-Reactive Recovery Technique (2)





FTTs Modelling (III): Based on UML (3)

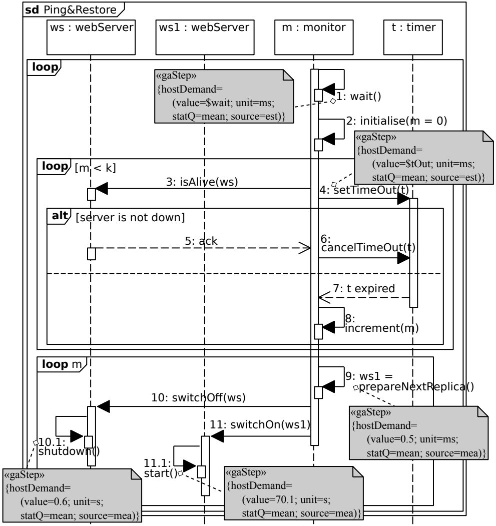
Switch Over Failing



- Spare replicas
- Attack threshold
- When threshold is reached → switch to another replica

FTTs Modelling (III): Based on UML (4)

Ping and Restore



- Spare replicas
- Alive replica?
- When not reply, switch to other replica

Performance Analysis (I): Some background. . . (1)

Petri nets subclasses

- State Machine: $\forall t \in T, |t^\bullet| = |^\bullet t| = 1$
- Marked Graph: $\forall p \in P, |^\bullet p| = |p^\bullet| = 1$

Performance Analysis (I): Some background. . . (1)

Petri nets subclasses

- State Machine: $\forall t \in T, |t^{\bullet}| = |{}^{\bullet}t| = 1$
- Marked Graph: $\forall p \in P, |{}^{\bullet}p| = |p^{\bullet}| = 1$

Process Petri nets (PPNs): a (really) interesting subclass

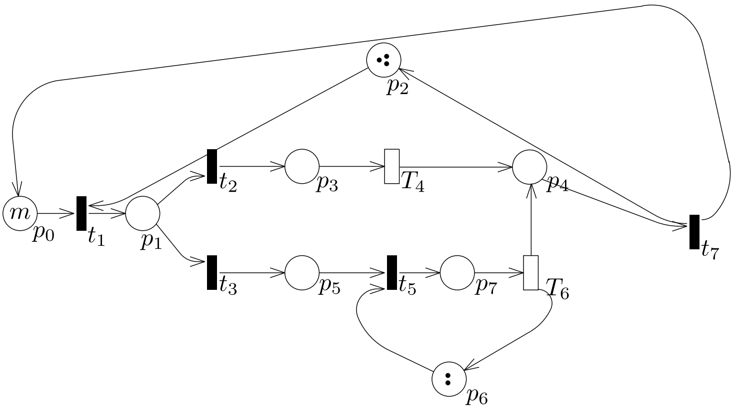
- Shared resources systems
- Any process which involves resource usage to complete
- Different jobs with dissimilar handling
- Examples: Assembly lines, Service-Oriented-Architecture systems. . .

Tricas F., Deadlock Analysis, Prevention and Avoidance in Sequential Resource Allocation Systems.

Ph.D. Dissertation, Universidad de Zaragoza, 2003

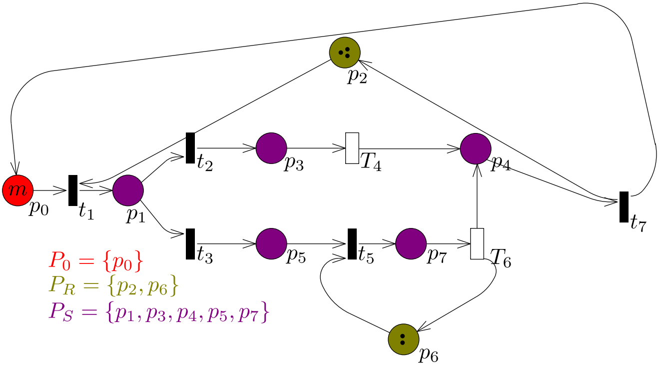
Performance Analysis (I): Some background. . . (2)

Informal definition of a PPN – an example



Performance Analysis (I): Some background... (2)

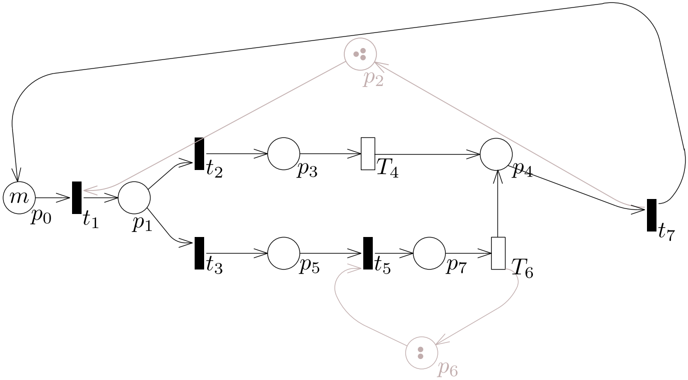
Informal definition of a PPN – an example



- Places can be divided in three subsets: $P = P_0 \cup P_S \cup P_R$
  - *Process-idle place*, $P_0 = \{p_0\}$
  - *Process-activity places*, $P_S \neq \emptyset, P_S \cap P_0 = \emptyset, P_S \cap P_R = \emptyset$
  - *Resources places*, $P_R = \{r_1, \ldots, r_n\},\ n > 0, P_R \cap P_0 = \emptyset$

Performance Analysis (I): Some background. . . (2)

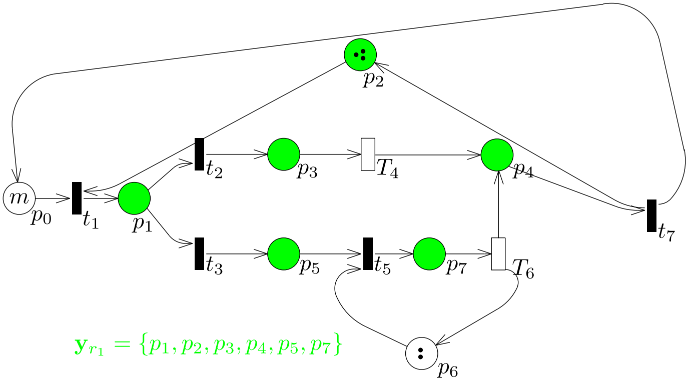
Informal definition of a PPN – an example



- When removing P_R places, we get a strongly connected state machine, s.t. every cycle contains p_0

Performance Analysis (I): Some background... (2)

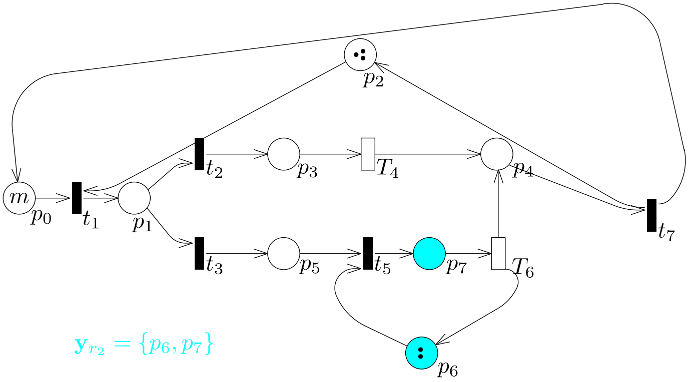
Informal definition of a PPN – an example



- For each $r \in P_R$, there exists a unique minimal p-semiflow associated to $r, \mathbf{y}_r \in \mathbb{N}^{|P|}$ s.t. it contains on its support just the resource r and does not contains p_0

Performance Analysis (I): Some background. . . (2)

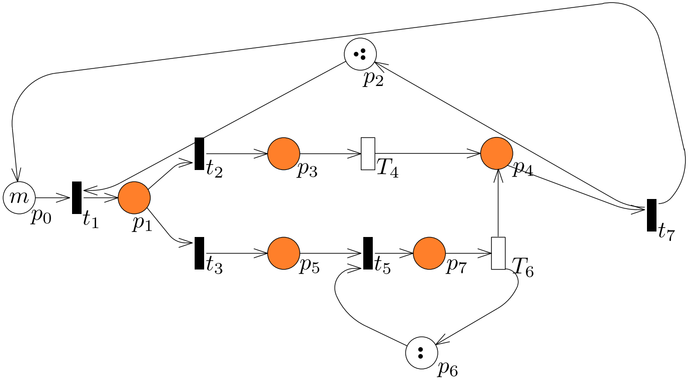
Informal definition of a PPN – an example



- For each $r \in P_R$, there exists a unique minimal p-semiflow associated to $r, \mathbf{y}_r \in \mathbb{N}^{|P|}$ s.t. it contains on its support just the resource r and does not contains p_0

Performance Analysis (I): Some background. . . (2)

Informal definition of a PPN – an example



- Activity places set P_S does not contain neither resource places, nor process-idle place

Performance Analysis (I): Some background. . . (3)

A LPP for a lower bound for the *average inter-firing time* of transition t_i

- Little's law: $L = \lambda \cdot W$ (queue length, arrival rate, waiting time)

Performance Analysis (I): Some background. . . (3)

A LPP for a lower bound for the *average inter-firing time* of transition t_i

- Little's law: $L = \lambda \cdot W$ (queue length, arrival rate, waiting time)
- $\overline{\mathbf{m}} \geq \mathbf{Pre} \cdot \mathbf{D} \cdot \Theta$

Performance Analysis (I): Some background. . . (3)

A LPP for a lower bound for the *average inter-firing time* of transition t_i

- Little's law: $L = \lambda \cdot W$ (queue length, arrival rate, waiting time)
- $\overline{\mathbf{m}} \geq \mathbf{Pre} \cdot \mathbf{D} \cdot \Theta$

$$\begin{aligned} &\textit{Maximize } \Theta : \\ &\qquad \overline{\mathbf{m}} \geq \mathbf{Pre} \cdot \mathbf{D} \cdot \Theta \\ &\qquad \overline{\mathbf{m}} = \mathbf{m_0} + \mathbf{C} \cdot \sigma \\ &\qquad \sigma \geq 0 \end{aligned}$$

Performance Analysis (I): Some background. . . (3)

A LPP for a lower bound for the *average inter-firing time* of transition t_i

- Little's law: $L = \lambda \cdot W$ (queue length, arrival rate, waiting time)
- $\overline{\mathbf{m}} \geq \mathbf{Pre} \cdot \mathbf{D} \cdot \Theta$

$$\begin{aligned} & \textit{Maximize } \Theta : \\ & \qquad \overline{\mathbf{m}} \geq \mathbf{Pre} \cdot \mathbf{D} \cdot \Theta \\ & \qquad \overline{\mathbf{m}} = \mathbf{m_0} + \mathbf{C} \cdot \sigma \\ & \qquad \sigma \geq 0 \end{aligned}$$

$$\begin{aligned} \textit{maximum } & \mathbf{y} \cdot \mathbf{Pre} \cdot \mathbf{D} \\ \textit{subject to } & \mathbf{y} \cdot \mathbf{C} = \mathbf{0} \\ & \mathbf{y} \cdot \mathbf{m_0} = 1 \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

- $\mathbf{y}$ is the slowest p-semiflow of the system (bottleneck)
- Our aim: Find next constraining p-semiflow

Campos, J. & Silva, M., Structural Techniques and Performance Bounds of Stochastic Petri Net Models. *Advances in PNs*, 1992

Performance Analysis (II): SMGs Regrowing Strategy (1)

Algorithm 1 in the Dissertation

- Input data: SMG, accuracy
- Output data: sharp performance bound, bottleneck

Performance Analysis (II): SMGs Regrowing Strategy (1)

Algorithm 1 in the Dissertation

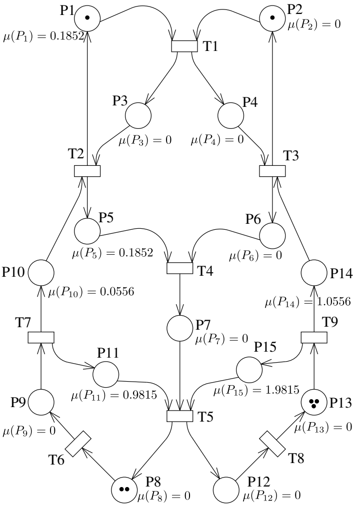
- Input data: SMG, accuracy
- Output data: sharp performance bound, bottleneck

Algorithm steps

1. Calculate initial upper throughput bound and initial bottleneck cycle
2. Calculate tight marking and slacks
   - Computed by a LPP (Carmona et al., Scheduling Synchronous Elastic Designs. *ACSD*, 2009)
   - The lower the slack, the higher the probability that place will constrain
3. Iterate until no significant improvement is achieved
   1. Look for place with minimum slack and add it
   2. Calculate new throughput bound

Performance Analysis (II): SMGs Regrowing Strategy (2)

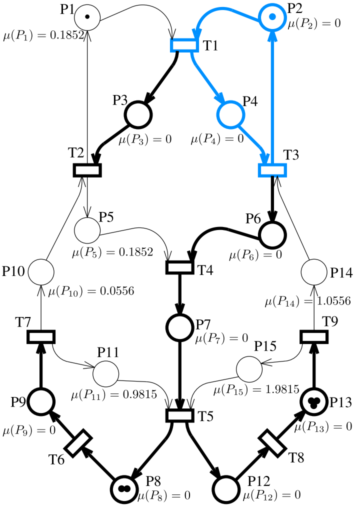
A running example for easy understanding



Consider an $\epsilon = 0.01$

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

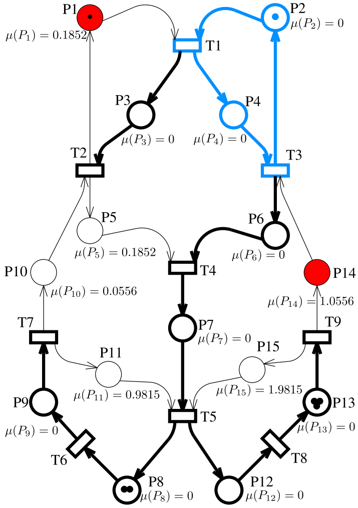


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	-	-	0.3704	-	-

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

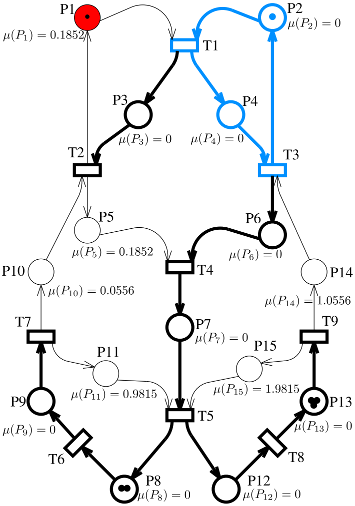


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	-	0.3704	-	-

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

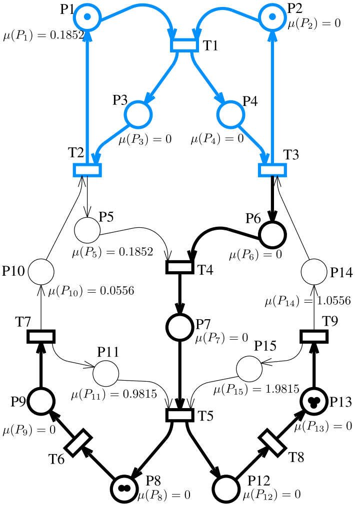


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1 , p_{14}	p_1	0.3704	-	-

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

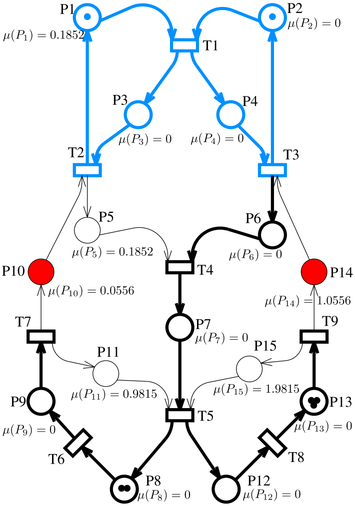


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1 , p_{14}	p_1	0.3704	-	-
1	-	-	0.322581	12.9%	12.9%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

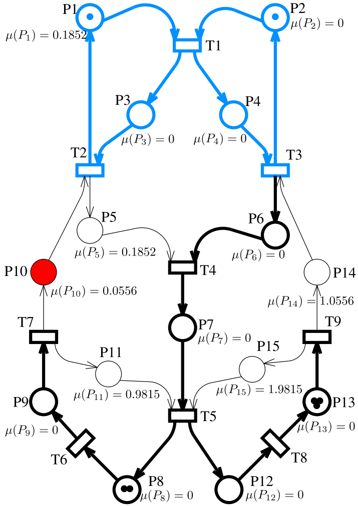


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	p_1	0.3704	-	-
1	p_{10}, p_{14}	-	0.322581	12.9%	12.9%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

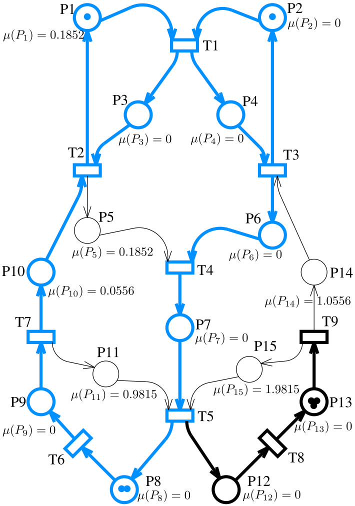


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1 , p_{14}	p_1	0.3704	-	-
1	p_{10} , p_{14}	p_{10}	0.322581	12.9%	12.9%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

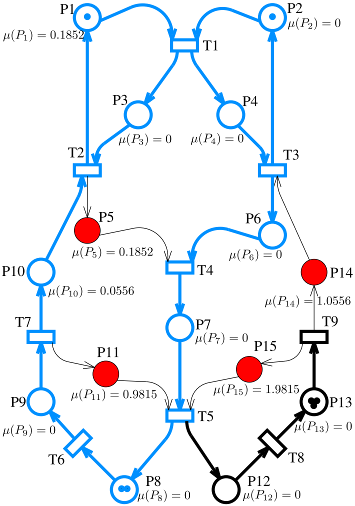


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	p_1	0.3704	-	-
1	p_{10}, p_{14}	p_{10}	0.322581	12.9%	12.9%
2	-	-	0.297914	7.647%	19.563%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

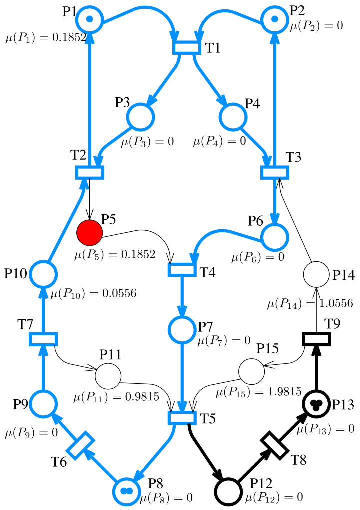


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	p_1	0.3704	-	-
1	p_{10}, p_{14}	p_{10}	0.322581	12.9%	12.9%
2	$p_5, p_{11}, p_{14}, p_{15}$	-	0.297914	7.647%	19.563%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

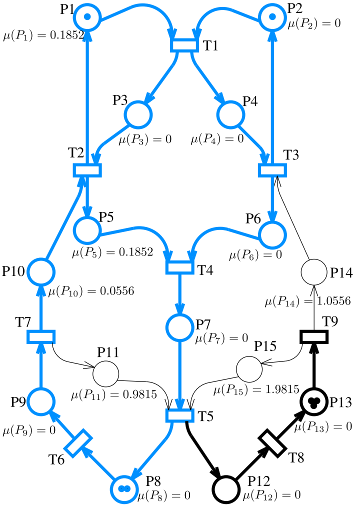


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	p_1	0.3704	-	-
1	p_{10}, p_{14}	p_{10}	0.322581	12.9%	12.9%
2	$p_5, p_{11}, p_{14}, p_{15}$	p_5	0.297914	7.647%	19.563%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

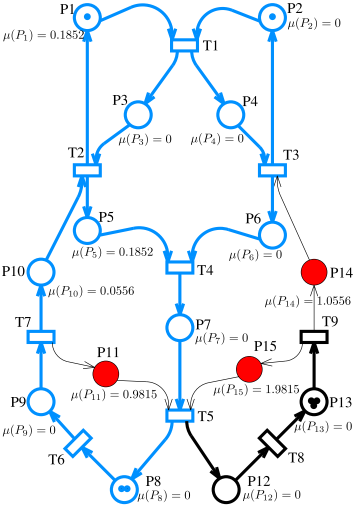


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	p_1	0.3704	-	-
1	p_{10}, p_{14}	p_{10}	0.322581	12.91%	12.91%
2	p_5, p_{11} p_{14}, p_{15}	p_5	0.297914	7.647%	19.563%
3	-	-	0.288401	3.193%	22.137%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

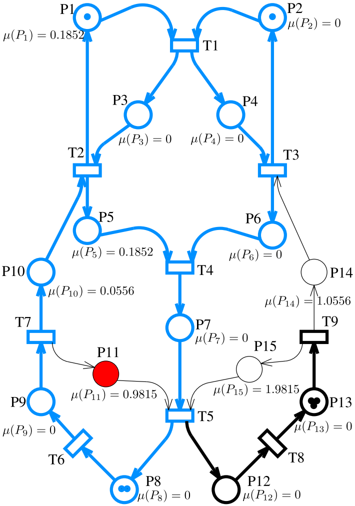


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	p_1	0.3704	-	-
1	p_{10}, p_{14}	p_{10}	0.322581	12.9%	12.9%
2	$p_5, p_{11}, p_{14}, p_{15}$	p_5	0.297914	7.647%	19.563%
3	p_{11}, p_{14}, p_{15}	-	0.288401	3.193%	22.137%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding

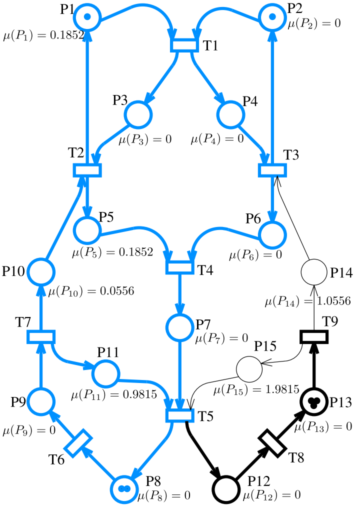


Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	p_1	0.3704	-	-
1	p_{10}, p_{14}	p_{10}	0.322581	12.9%	12.9%
2	$p_5, p_{11}, p_{14}, p_{15}$	p_5	0.297914	7.647%	19.563%
3	p_{11}, p_{14}, p_{15}	p_{11}	0.288401	3.193%	22.137%

Performance Analysis (II): SMGs Regrowing Strategy (2)

A running example for easy understanding



Consider an $\epsilon = 0.01$

Iteration step	Candidates places	Added	Θ	$\%_{last}$	$\%_{initial}$
0	p_1, p_{14}	p_1	0.3704	-	-
1	p_{10}, p_{14}	p_{10}	0.322581	12.9%	12.9%
2	p_5, p_{11}, p_{14}, p_{15}	p_5	0.297914	7.647%	19.563%
3	p_{11}, p_{14}, p_{15}	p_{11}	0.288401	3.193%	22.137%
4	-	-	0.288401	0%	22.137%

Performance Analysis (II): SMGs Regrowing Strategy (3)

Experiments and Discussion (1)

Benchmarking and used tools

- ISCAS benchmark
  - Strongly connected components of the ISCAS graphs
  - Initial marking randomly selected in $[1 \ldots 10]$
  - Delay of transitions randomly selected in $[0.1 \ldots 1]$
- Strategy implemented in MATLAB (linprog)
- Simulation tool: GreatSPN
  - Confidence level 99%; accuracy 1%
- Host: Pentium IV 3.6GHz, 2GB DDR2 533MHz RAM

Performance Analysis (II): SMGs Regrowing Strategy (3)

Experiments and Discussion (2)

Graph	Size		% Size		Regrowing steps	Initial thr. bound	Θ
	$\lvert P \rvert$	$\lvert T \rvert$	$\lvert P' \rvert$ (%)	$\lvert T' \rvert$ (%)			
s1423	1107	792	79 (7.13%)	76 (9.59%)	3	0.236010	0.235213 (0.34%)
s1488	1567	1128	91 (5.8%)	86 (7.62%)	6	0.201300	0.173127 (13.99%)
s208	27	24	27 (100%)	24 (100%)	3	0.409390	0.377683 (7.75%)
s27	54	44	19 (35.18%)	18 (40.9%)	1	0.305960	0.304987 (0.31%)
s349	187	146	26 (13.9%)	24 (16.44%)	2	0.340320	0.327867 (3.66%)
s444	92	68	14 (15.21%)	12 (17.64%)	2	0.181670	0.181260 (0.22%)
s510	1038	734	45 (4.33%)	40 (5.45%)	5	0.133030	0.117819 (11.43%)
s526	113	92	18 (15.93%)	16 (17.39%)	2	0.313490	0.305860 (2.43%)
s713	271	208	11 (4.06%)	10 (4.8%)	1	0.428720	0.427840 (0.2%)
s820	1162	848	40 (3.44%)	38 (4.48%)	2	0.161060	0.147483 (8.43%)
s832	1293	948	84 (6.5%)	78 (12.04%)	5	0.239429	0.208798 (12.79%)
s953	415	312	88 (11.36%)	82 (26.28%)	6	0.369214	0.337811 (8.50%)

- Sharp upper bound in few regrowing steps
  - Improvement varies from 0.2% to 14%
- Uses a very low percentage of the size of the original graph
  - Lower than 10% (in most of cases)

Performance Analysis (II): SMGs Regrowing Strategy (3)

Experiments and Discussion (2)

Graph	Size		% Size		Regrowing steps	Initial thr. bound	Θ
	$\lvert P\rvert$	$\lvert T\rvert$	$\lvert P'\rvert$ (%)	$\lvert T'\rvert$ (%)			
s1423	1107	792	79 (7.13%)	76 (9.59%)	3	0.236010	0.235213 (0.34%)
s1488	1567	1128	91 (5.8%)	86 (7.62%)	6	0.201300	0.173127 (13.99%)
s208	27	24	27 (100%)	24 (100%)	3	0.409390	0.377683 (7.75%)
s27	54	44	19 (35.18%)	18 (40.9%)	1	0.305960	0.304987 (0.31%)
s349	187	146	26 (13.9%)	24 (16.44%)	2	0.340320	0.327867 (3.66%)
s444	92	68	14 (15.21%)	12 (17.64%)	2	0.181670	0.181260 (0.22%)
s510	1038	734	45 (4.33%)	40 (5.45%)	5	0.133030	0.117819 (11.43%)
s526	113	92	18 (15.93%)	16 (17.39%)	2	0.313490	0.305860 (2.43%)
s713	271	208	11 (4.06%)	10 (4.8%)	1	0.428720	0.427840 (0.2%)
s820	1162	848	40 (3.44%)	38 (4.48%)	2	0.161060	0.147483 (8.43%)
s832	1293	948	84 (6.5%)	78 (12.04%)	5	0.239429	0.208798 (12.79%)
s953	415	312	88 (11.36%)	82 (26.28%)	6	0.369214	0.337811 (8.50%)

- Sharp upper bound in few regrowing steps
  - Improvement varies from 0.2% to 14%
- Uses a very low percentage of the size of the original graph
  - Lower than 10% (in most of cases)

Performance Analysis (II): SMGs Regrowing Strategy (3)

Experiments and Discussion (3)

Graph	Original graph thr. CPU time (s)	Θ CPU time (s)	Original graph thr.	Θ	% thr.
s1423	59948.980	8.283	0.222720	0.235270	5.63%
s1488	36717.156	7.165	0.168760	0.172154	2.01%
s208	0.492	0.492	0.376892	0.376892	0%
s27	2166.002	0.954	0.305082	0.306166	0.35%
s349	141.210	0.441	0.328340	0.327398	−0.28%
s444	2278.231	0.205	0.181069	0.181260	0.11%
s510	13669.814	1.358	0.117500	0.118040	0.46%
s526	129.181	0.344	0.270010	0.305860	13.27%
s713	628.503	0.405	0.411630	0.427840	3.94%
s820	20775.811	0.788	0.144770	0.147699	2.02%
s832	16165.863	1.914	0.196920	0.208873	6.07%
s953	453.850	19.155	0.327910	0.338644	3.27%

- Θ CPU time insignificant respect to original thr CPU time
- Improvement varies from very close value to 13% over the real thr
  - Slow cycles far away from critical cycle?
- Negative relative error caused by simulation confidence interval

Performance Analysis (II): SMGs Regrowing Strategy (3)

Experiments and Discussion (3)

Graph	Original graph thr. CPU time (s)	Θ CPU time (s)	Original graph thr.	Θ	% thr.
s1423	59948.980	8.283	0.222720	0.235270	5.63%
s1488	36717.156	7.165	0.168760	0.172154	2.01%
s208	0.492	0.492	0.376892	0.376892	0%
s27	2166.002	0.954	0.305082	0.306166	0.35%
s349	141.210	0.441	0.328340	0.327398	−0.28%
s444	2278.231	0.205	0.181069	0.181260	0.11%
s510	13669.814	1.358	0.117500	0.118040	0.46%
s526	129.181	0.344	0.270010	0.305860	13.27%
s713	628.503	0.405	0.411630	0.427840	3.94%
s820	20775.811	0.788	0.144770	0.147699	2.02%
s832	16165.863	1.914	0.196920	0.208873	6.07%
s953	453.850	19.155	0.327910	0.338644	3.27%

- Θ CPU time insignificant respect to original thr CPU time
- Improvement varies from very close value to 13% over the real thr
  - Slow cycles far away from critical cycle?
- Negative relative error caused by simulation confidence interval

Performance Analysis (III): PPNs Regrowing Strategy (1)

Which is the next p-semiflow constraining?

Generalising from SMGs to PPNs

- Previous method not longer valid
- Need other technique for next p-semiflow more likely to constrain

Performance Analysis (III): PPNs Regrowing Strategy (1)

Which is the next p-semiflow constraining?

Generalising from SMGs to PPNs

- Previous method not longer valid
- Need other technique for next p-semiflow more likely to constrain

$$
\begin{aligned}
\textit{maximum } & \mathbf{y}' \cdot \mathbf{Pre} \cdot \mathbf{D} \\
\textit{subject to } & \mathbf{y}' \cdot \mathbf{C} = \mathbf{0} \\
& \mathbf{y}' \cdot \mathbf{m_0} = 1 \\
& \mathbf{y}'(p) \geq h, \ \forall p \in \mathbf{Q} \\
& \sum_{p \in V} \mathbf{y}'(p) \geq h \\
\textit{where } & V = \{v | v \in {}^\bullet(\|\mathbf{y}^*\|^\bullet) \setminus \|\mathbf{y}^*\|\} \\
& Q = \{q \in P, q \in \|\mathbf{y}^*\|\}
\end{aligned}
$$

- Value h strictly positive and arbitrarily small
- Avoid numerical problems ($h = 0$) and LP feasible

Performance Analysis (III): PPNs Regrowing Strategy (1)

Which is the next p-semiflow constraining?

Generalising from SMGs to PPNs

- Previous method not longer valid
- Need other technique for next p-semiflow more likely to constrain

$$
\begin{aligned}
\textit{maximum}\ & \mathbf{y}' \cdot \mathbf{Pre} \cdot \mathbf{D} \\
\textit{subject to}\ & \mathbf{y}' \cdot \mathbf{C} = \mathbf{0} \\
& \mathbf{y}' \cdot \mathbf{m_0} = 1 \\
& \mathbf{y}'(p) \geq h,\ \forall p \in \mathbf{Q} \\
& \sum_{p \in V} \mathbf{y}'(p) \geq h
\end{aligned}
$$

$$
\begin{aligned}
\textit{where}\ V &= \{v | v \in {}^{\bullet}(\|\mathbf{y}^*\|^{\bullet}) \setminus \|\mathbf{y}^*\|\} \\
Q &= \{q \in P, q \in \|\mathbf{y}^*\|\}
\end{aligned}
$$

- Value h strictly positive and arbitrarily small
- Avoid numerical problems ($h = 0$) and LP feasible

$$
\begin{aligned}
\textit{maximum}\ & h \\
\textit{subject to}\ & \mathbf{y} \cdot \mathbf{C} = \mathbf{0} \\
& \mathbf{y} \cdot \mathbf{m_0} = 1 \\
& \mathbf{y} \geq h \cdot \mathbf{1} \\
& h > 0
\end{aligned}
$$

Performance Analysis (III): PPNs Regrowing Strategy (2)

Algorithm 2 in the Dissertation

- Input data: PPN, accuracy
- Output data: sharp performance bound, bottleneck

Performance Analysis (III): PPNs Regrowing Strategy (2)

Algorithm 2 in the Dissertation

- Input data: PPN, accuracy
- Output data: sharp performance bound, bottleneck

Algorithm steps

1. Calculate initial upper throughput bound and initial bottleneck cycle
2. Calculate value h
3. Iterate until no significant improvement is achieved or all places are considered
    1. Compute the next constraining p-semiflow
    2. Calculate new thr. bound

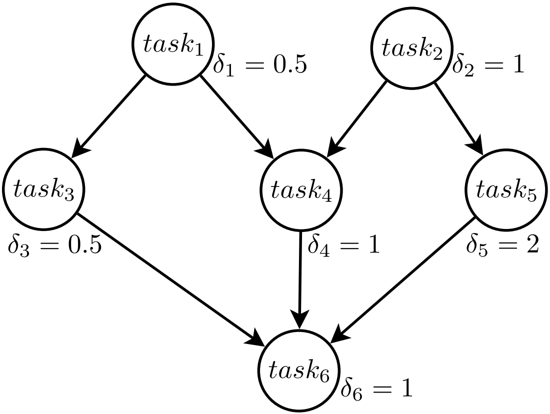
$$
\begin{aligned}
\textit{maximum } & \mathbf{y}' \cdot \mathbf{Pre} \cdot \mathbf{D} \\
\textit{subject to } & \mathbf{y}' \cdot \mathbf{C} = \mathbf{0} \\
& \mathbf{y}' \cdot \mathbf{m_0} = 1 \\
& \mathbf{y}'(p) \geq h, \ \forall p \in \mathbf{Q} \\
& \sum_{p \in V} \mathbf{y}'(p) \geq h
\end{aligned}
$$

$$\textit{where } V = \{v | v \in {}^{\bullet}(\|\mathbf{y}^*\|^{\bullet}) \setminus \|\mathbf{y}^*\|\}$$

$$Q = \{q \in P, q \in \|\mathbf{y}^*\|\}$$

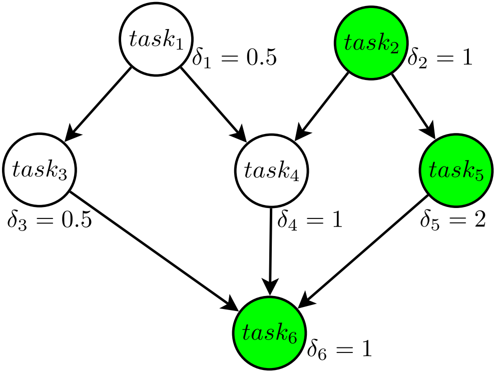
Data-Throttling in Scientific Workflows (I)

An intuitive idea: The slack concept



Data-Throttling in Scientific Workflows (I)

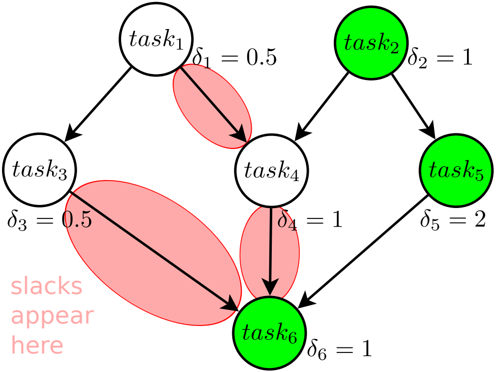
An intuitive idea: The slack concept



δ execution time; $tx = 1$ transmission time

Data-Throttling in Scientific Workflows (I)

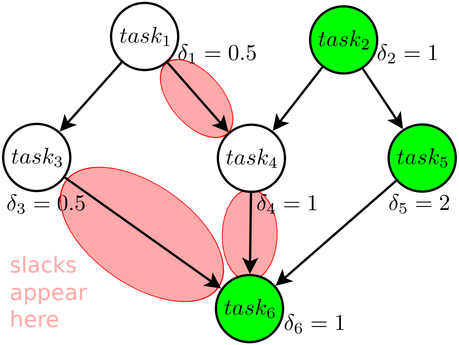
An intuitive idea: The slack concept



δ execution time; $tx = 1$ transmission time

Data-Throttling in Scientific Workflows (I)

An intuitive idea: The slack concept



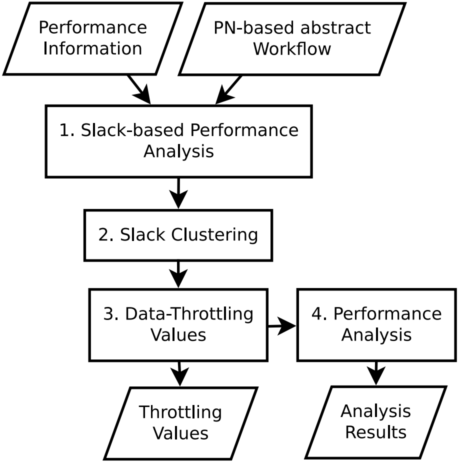
δ execution time; $tx = 1$ transmission time

Assume $tx = 1$

- Slowest path:
  $$\Theta = \frac{1}{\sum(\delta + tx)} = \frac{1}{6}$$
  (Θ: inverse of execution time of slowest path)
- Slack: how much faster is a path w.r.t. another until a synchronisation point
  - $\mu_{1,4} = \frac{2 - 1.5}{6} = \frac{0.5}{6}$
  - $\mu_{4,6} = \frac{5 - 3.5}{6} = \frac{1.5}{6}$
  - $\mu_{3,6} = \frac{5 - 3}{6} = \frac{1}{3}$

Data-Throttling in Scientific Workflows (II)

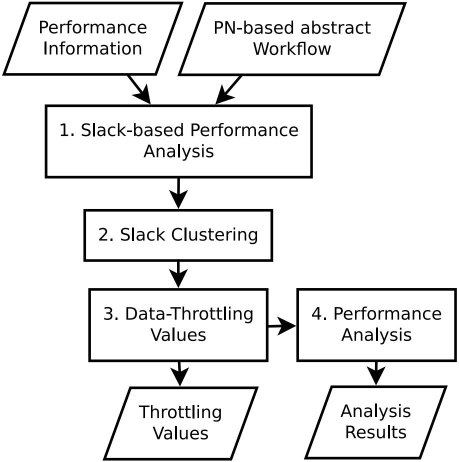
Automating Data-Throttling Analysis: An algorithm



- Inputs: Performance estimation (i.e., DAX annotations) + PN-based model
- Outputs: Data-throttling values + Analysis results

Data-Throttling in Scientific Workflows (II)

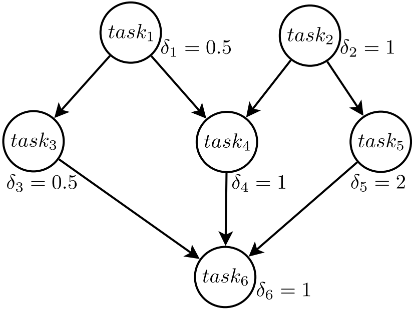
Automating Data-Throttling Analysis: An algorithm



- Inputs: Performance estimation (i.e., DAX annotations) + PN-based model
- Outputs: Data-throttling values + Analysis results
- 4 steps
  1. Compute slack values
  2. Cluster slacks
  3. Compute data-throttling values
  4. Performance analysis
     - With and w/out data-throttling

Data-Throttling in Scientific Workflows (III)

Applying the algorithm to an example



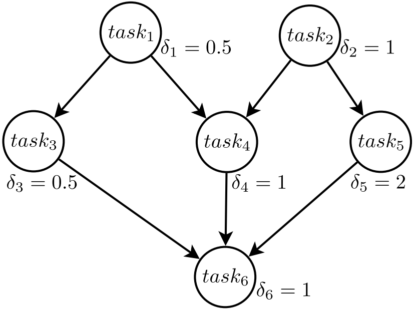
Recall: slacks on synchronisation points

Assumptions

- Bandwidth=100Mbps
- Data-sets equal to 10MiB
- Dedicated network topology

Data-Throttling in Scientific Workflows (III)

Applying the algorithm to an example



Recall: slacks on synchronisation points

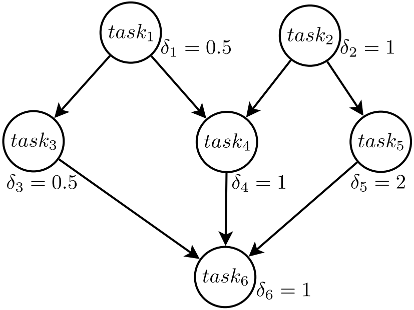
Assumptions

- Bandwidth=100Mbps
- Data-sets equal to 10MiB
- Dedicated network topology

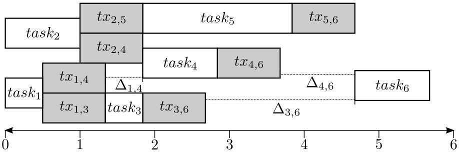
- Slowest path: $2 \rightarrow 5 \rightarrow 6$
- Slacks: $\mu_{1,4}, \mu_{3,6}, \mu_{4,6}$

Data-Throttling in Scientific Workflows (III)

Applying the algorithm to an example



Recall: slacks on synchronisation points



Makespan: 5.6779 seconds

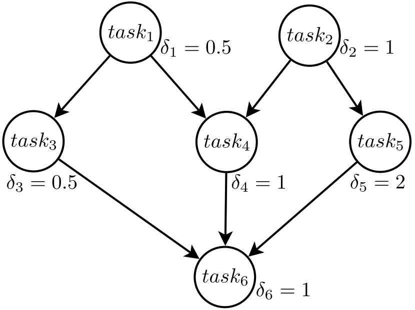
Assumptions

- Bandwidth=100Mbps
- Data-sets equal to 10MiB
- Dedicated network topology

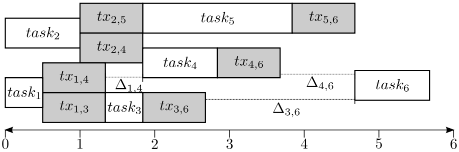
- Slowest path: $2 \rightarrow 5 \rightarrow 6$
- Slacks: $\mu_{1,4}, \mu_{3,6}, \mu_{4,6}$

Data-Throttling in Scientific Workflows (III)

Applying the algorithm to an example



Recall: slacks on synchronisation points



Makespan: 5.6779 seconds

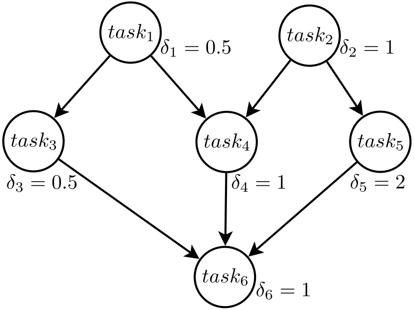
Assumptions

- Bandwidth=100Mbps
- Data-sets equal to 10MiB
- Dedicated network topology

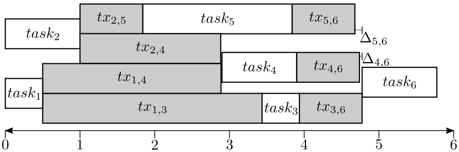
- Slowest path: $2 \rightarrow 5 \rightarrow 6$
- Slacks: $\mu_{1,4}, \mu_{3,6}, \mu_{4,6}$
  - $1 \rightarrow 4$ adjust to 28.57%
  - $3 \rightarrow 6$ adjust to 35.15%
  - $4 \rightarrow 6$ adjust to 44.55%

Data-Throttling in Scientific Workflows (III)

Applying the algorithm to an example



Recall: slacks on synchronisation points



Makespan: 5.7750 seconds

(negligent impact on makespan BUT input buffer waiting times reduced)

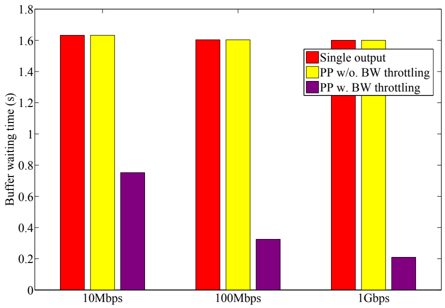
Assumptions

- Bandwidth=100Mbps
- Data-sets equal to 10MiB
- Dedicated network topology

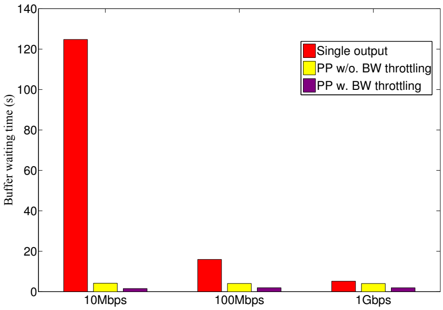
- Slowest path: $2 \rightarrow 5 \rightarrow 6$
- Slacks: $\mu_{1,4}, \mu_{3,6}, \mu_{4,6}$
  - $1 \rightarrow 4$ adjust to 28.57%
  - $3 \rightarrow 6$ adjust to 35.15%
  - $4 \rightarrow 6$ adjust to 44.55%

Data-Throttling in Scientific Workflows (IV)

Input Buffers Usage – some plots of Montage workflow tasks



(a) Workflow task *mImgTbl*



(b) Workflow task *mConcatFit*

- Data-throttling has great impact on input buffers
- Outperforms both other topologies

Resource Optimisation (I)

- Recall: FT systems with shared resources
  - Can we add more resources to compensate throughput degradation?

Resource Optimisation (I)

- Recall: FT systems with shared resources
  - Can we add more resources to compensate throughput degradation?
- Yes, we can (some restrictions apply. . .)

Resource Optimisation (I)

- Recall: FT systems with shared resources
  - Can we add more resources to compensate throughput degradation?
- Yes, we can (some restrictions apply. . .)

Considered approaches

- Budget limited while compensating a throughput degradation
- Budget unlimited while maintaining a given throughput

Resource Optimisation (II): A heuristics

Algorithm 3 in the Dissertation – budget limited

- Input data: PPN, budget, resource costs
- Output data: Resource increments

Resource Optimisation (II): A heuristics

Algorithm 3 in the Dissertation – budget limited

- Input data: PPN, budget, resource costs
- Output data: Resource increments

Algorithm steps

1. Initial bottleneck
2. Iterate until no enough budget or next resource is process-idle place
   - Compute next constraining p-semiflow, and its associated resource (LPP in right column)
   - How much is this new increment?
3. When all resources are incremented and still enough budget $\rightarrow$ assign it

$$
\begin{aligned}
&\textit{minimum} \sum_{j=1}^{k} \alpha_j \\
&\textit{subject to } \mathbf{y_{k+1}} \cdot \mathbf{Pre} \cdot \mathbf{D} = \mathbf{y_1} \cdot \mathbf{Pre} \cdot \mathbf{D} \\
&\quad \mathbf{y_{k+1}} \cdot \mathbf{C} = \mathbf{0} \\
&\quad \mathbf{y_{k+1}} \cdot \mathbf{m_0^{\Delta}} = \mathbf{y_j} \cdot \mathbf{m_0^{\Delta}}, \ \forall j \in \{1 \ldots k\} \\
&\quad \mathbf{m_0}^{\Delta} = \begin{cases} \mathbf{m_0}(p) + \alpha_j, & p \in \mathbf{A} \\ \mathbf{m_0}(p), & \textit{otherwise} \end{cases} \\
&\quad \mathbf{y_{k+1}}(p) = 0, \ p \in \mathbf{A} \\
&\quad \mathbf{y_{k+1}}, \ \alpha_j \geq 0, \ \forall j \in \{1 \ldots k\}
\end{aligned}
$$

Resource Optimisation (III): Initial marking maintaining thr.

Algorithm 4 in the dissertation – budget unlimited

- Input data: PPN with some FT added, Θ, Set of p-semiflows added/modified when adding FTs
- Output data: Resource increments

Resource Optimisation (III): Initial marking maintaining thr.

Algorithm 4 in the dissertation – budget unlimited

- Input data: PPN with some FT added, Θ, Set of p-semiflows added/modified when adding FTs
- Output data: Resource increments

Algorithm steps

1. Iterate for each p-semiflow modified/added by the FT technique
    - Get maximum between current initial resource instances and the resources needed for maintaining giving Θ

Resource Optimisation (IV): Guaranteeing minimum cost

$$\begin{array}{rcl} \textit{minimum} & & \left(\sum_{i=1}^{n} c_i^r \cdot \alpha_i + \sum_{j=1}^{m} \left(c_j^d \cdot \beta_j^d + c_j^c \cdot \beta_j^c + c_j^{rm} \cdot \beta_j^{rm}\right)\right) \textit{ subject to} \\ \mathbf{m_0}(r_i) + \alpha_i & \geq & \Theta \cdot \mathbf{y}_i \cdot \mathbf{Pre} \cdot \mathbf{D}' \\ \delta'(T^j_{detect}) & = & \delta(T^j_{detect}) - \beta_j^d \\ \delta'(T^j_c) & = & \delta(T^j_c) - \beta_j^c \\ \delta'(T^j_{rec}) & = & \delta(T^j_{rec}) - \beta_j^{rm} \\ \delta'(t) & \geq & \delta_{min}(t), \forall t \in T \\ \alpha_i, \beta_j^d, \beta_j^c, \beta_j^{rm} & \geq & 0, \alpha_i \in \mathbb{N}, \forall i \in [1 \ldots n], \forall j \in [1 \ldots m] \end{array} \quad (1)$$

Integer-Linear Programming Problem (ILPP)

- Minimum cost that guarantees a compensation of the throughput

Resource Optimisation (IV): Guaranteeing minimum cost

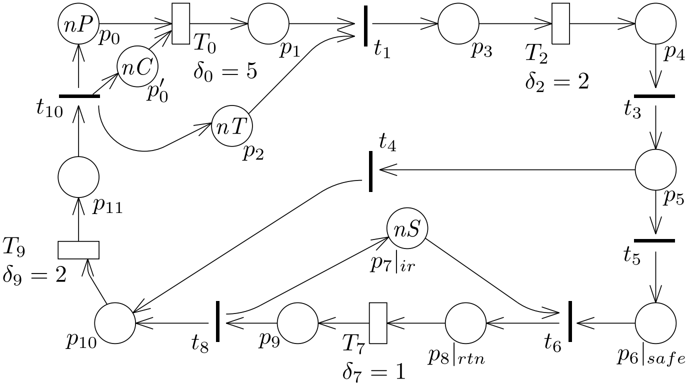
$$
\begin{array}{rcl}
\textit{minimum} & & \left(\sum_{i=1}^{n} c_i^r \cdot \alpha_i + \sum_{j=1}^{m} \left(c_j^d \cdot \beta_j^d + c_j^c \cdot \beta_j^c + c_j^{rm} \cdot \beta_j^{rm} \right) \right) \textit{ subject to} \\
\mathbf{m_0}(r_i) + \alpha_i & \geq & \Theta \cdot \mathbf{y}_i \cdot \mathbf{Pre} \cdot \mathbf{D}' \\
\delta'(T^j_{detect}) & = & \delta(T^j_{detect}) - \beta^d_j \\
\delta'(T^j_c) & = & \delta(T^j_c) - \beta^c_j \\
\delta'(T^j_{rec}) & = & \delta(T^j_{rec}) - \beta^{rm}_j \\
\delta'(t) & \geq & \delta_{min}(t), \forall t \in T \\
\alpha_i, \beta^d_j, \beta^c_j, \beta^{rm}_j & \geq & 0, \alpha_i \in \mathbb{N}, \forall i \in [1 \ldots n], \forall j \in [1 \ldots m]
\end{array}
\quad (1)
$$

Integer-Linear Programming Problem (ILPP)

- Minimum cost that guarantees a compensation of the throughput
- Why an ILPP instead of a LPP?
  - Resource instances are natural numbers

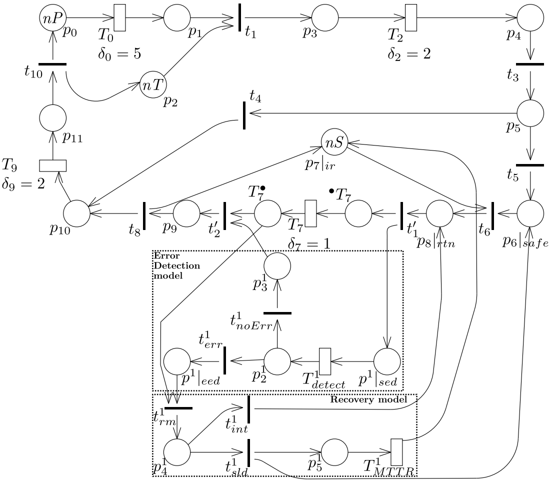
Case Study (I): Problem Description (1)

A packet-routing algorithm



Case Study (I): Problem Description (2)

A packet-routing algorithm: adding FT capabilities – reconfiguration with rollbackward



Case Study (II): Performance Analysis (1)

A packet-routing algorithm with FT capabilities – reconfiguration with rollbackward

- $nP = 10$, $nT = 2$ and $nS = 2$

Case Study (II): Performance Analysis (1)

A packet-routing algorithm with FT capabilities – reconfiguration with rollbackward

- $nP = 10$, $nT = 2$ and $nS = 2$
- Reconfiguration with rollbackward
  - 20% of errors, and 5% of solid faults

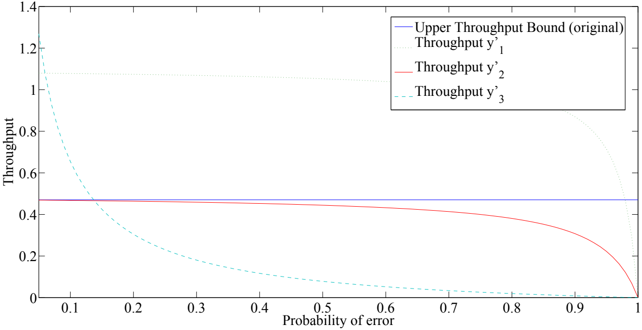
Case Study (II): Performance Analysis (1)

A packet-routing algorithm with FT capabilities – reconfiguration with rollbackward

- $nP = 10$, $nT = 2$ and $nS = 2$
- Reconfiguration with rollbackward
  - 20% of errors, and 5% of solid faults
- Before FT: $\mathbf{y} = \{p_2, p_3, p_4, p_5, p_6|_{safe}, p_8|_{rtn}, p_9, p_{10}, p_{11}\}$, $\Theta = 0.470588$
- After FT:
  - New PN semiflows are known
  - $\mathbf{y}'_1$, $\mathbf{y}'_2$ and $\mathbf{y}'_3$. $\Theta_1 = 1.073825$, $\Theta_2 = 0.463768$ and $\Theta_3 = 0.304762$
- New slowest: $\Theta' = 0.304762 \rightarrow$ a degradation of 35.23%

Case Study (II): Performance Analysis (2)

A packet-routing algorithm with FT capabilities: sensitive analysis



- Θ_1, Θ_2 and Θ_3 w.r.t. $r_e, r_e \in [0 \dots 1]$
- $\Theta_2 < \Theta$ for low probabilities of error
  - Holds until $r_e = 0.14$. From that point, Θ_3 becomes slowest
  - Moreover, it exponentially decreases
- $r_e \approx 0.8 \rightarrow \Theta_1, \Theta_2$ and Θ_3 quickly decrease and tend to zero

Case Study (III): Resource optimisation (1)

A packet-routing algorithm with FT capabilities: maintaining a given throughput

- $nP = 10$, $nT = 2$ and $nS = 2$

Case Study (III): Resource optimisation (1)

A packet-routing algorithm with FT capabilities: maintaining a given throughput

- $nP = 10$, $nT = 2$ and $nS = 2$
- Reconfiguration with rollbackward
  - 20% of errors, and 5% of solid faults

Case Study (III): Resource optimisation (1)

A packet-routing algorithm with FT capabilities: maintaining a given throughput

- $nP = 10$, $nT = 2$ and $nS = 2$
- Reconfiguration with rollbackward
  - 20% of errors, and 5% of solid faults
- Before FT: $\mathbf{y} = \{p_2, p_3, p_4, p_5, p_6|_{safe}, p_8|_{rtn}, p_9, p_{10}, p_{11}\}$, $\Theta = 0.470588$
- Interest in maintaining Θ while FT capabilities are present

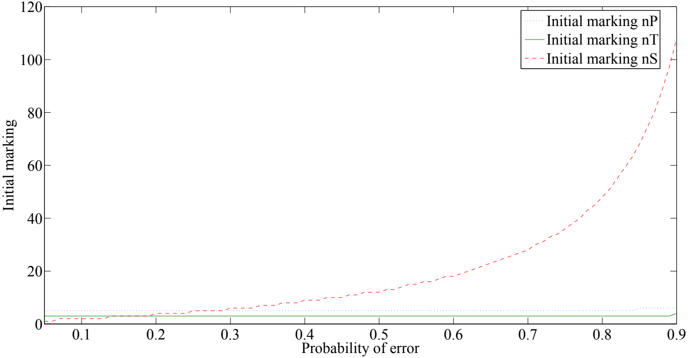
Case Study (III): Resource optimisation (1)

A packet-routing algorithm with FT capabilities: maintaining a given throughput

- $nP = 10$, $nT = 2$ and $nS = 2$
- Reconfiguration with rollbackward
  - 20% of errors, and 5% of solid faults
- Before FT: $\mathbf{y} = \{p_2, p_3, p_4, p_5, p_6|_{safe}, p_8|_{rtn}, p_9, p_{10}, p_{11}\}$, $\Theta = 0.470588$
- Interest in maintaining Θ while FT capabilities are present
- $\mathbf{m}'_\mathbf{0}(p_0) = \mathbf{m_0}(p_0) = 10, \mathbf{m}'_\mathbf{0}(p_2) = 3, \mathbf{m}'_\mathbf{0}(p_7) = 4$
  - Add another thread and two more filtering-threads to compensate a 20% of errors (and a 5% of them deriving in solid faults) using reconfiguration as FT technique

Case Study (III): Resource optimisation (2)

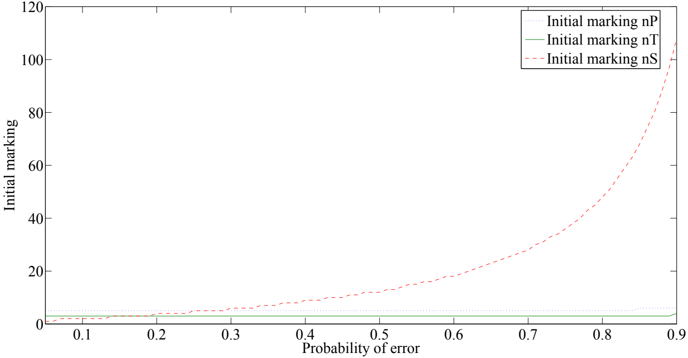
A packet-routing algorithm with FT capabilities: maintaining a given throughput



- Initial resources to maintain $\Theta = 0.470588$, $r_e \in [0 \ldots 1]$, steps of 0.01

Case Study (III): Resource optimisation (2)

A packet-routing algorithm with FT capabilities: maintaining a given throughput



- Initial resources to maintain $\Theta = 0.470588$, $r_e \in [0 \ldots 1]$, steps of 0.01
- No. packets and threads: remain equal
- No. filtering-threads: increases rapidly w.r.t. r_e

Conclusions (I)

- Security must be considered from the (early) beginning
- Be careful with the impact of adding security $\rightarrow$ trade-off analysis
- FT systems (sharing resources) modelled by PNs

Conclusions (I)

- Security must be considered from the (early) beginning
- Be careful with the impact of adding security → trade-off analysis
- FT systems (sharing resources) modelled by PNs

Contributions of this Dissertation – a Summary

- `SecAM`: A UML Security Analysis and Modelling profile

Conclusions (I)

- Security must be considered from the (early) beginning
- Be careful with the impact of adding security $\rightarrow$ trade-off analysis
- FT systems (sharing resources) modelled by PNs

Contributions of this Dissertation – a Summary

- SecAM: A UML Security Analysis and Modelling profile
- FTTs models (UML + PNs) helping to choose best suitable solution

Conclusions (I)

- Security must be considered from the (early) beginning
- Be careful with the impact of adding security $\rightarrow$ trade-off analysis
- FT systems (sharing resources) modelled by PNs

Contributions of this Dissertation – a Summary

- SecAM: A UML Security Analysis and Modelling profile
- FTTs models (UML + PNs) helping to choose best suitable solution
- Model-based methodology for performance prediction

Conclusions (I)

- Security must be considered from the (early) beginning
- Be careful with the impact of adding security $\rightarrow$ trade-off analysis
- FT systems (sharing resources) modelled by PNs

Contributions of this Dissertation – a Summary

- SecAM: A UML Security Analysis and Modelling profile
- FTTs models (UML + PNs) helping to choose best suitable solution
- Model-based methodology for performance prediction
- Performance estimation based on upper thr. bounds computation

Conclusions (I)

- Security must be considered from the (early) beginning
- Be careful with the impact of adding security → trade-off analysis
- FT systems (sharing resources) modelled by PNs

Contributions of this Dissertation – a Summary

- SecAM: A UML Security Analysis and Modelling profile
- FTTs models (UML + PNs) helping to choose best suitable solution
- Model-based methodology for performance prediction
- Performance estimation based on upper thr. bounds computation
- Data-throttling in scientific workflows: improvement of input buffer and network bandwidth usage

Conclusions (I)

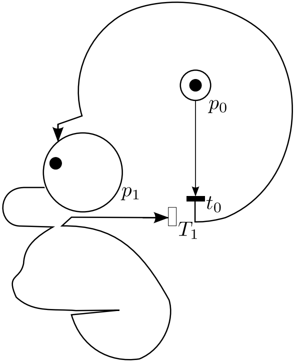
- Security must be considered from the (early) beginning
- Be careful with the impact of adding security → trade-off analysis
- FT systems (sharing resources) modelled by PNs

Contributions of this Dissertation – a Summary

- SecAM: A UML Security Analysis and Modelling profile
- FTTs models (UML + PNs) helping to choose best suitable solution
- Model-based methodology for performance prediction
- Performance estimation based on upper thr. bounds computation
- Data-throttling in scientific workflows: improvement of input buffer and network bandwidth usage
- Resource optimisation: increase no. resources while fulfilling requirements

Conclusions (II)

http://webdiis.unizar.es/GISED/?q=tool/peabrain

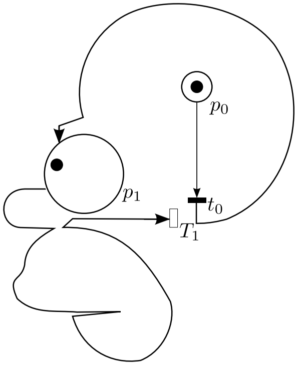


Contribution: the PeabraiN tool

- PeabraiN: a tool for performance estimation and resource optimisation computation in Petri nets
  - GPL v3
  - JAVA-based

Conclusions (II)

http://webdiis.unizar.es/GISED/?q=tool/peabrain



Contribution: the PeabraiN tool

- PeabraiN: a tool for performance estimation and resource optimisation computation in Petri nets
  - GPL v3
  - JAVA-based

Some stats

- 115 downloads (SF.net)
- Countries top three: Spain, France, Brazil

Conclusions (III): Published Papers (1)

Total numbers

- Two papers on JCR-indexed journals (SMC, JRR)
- Seven papers on international peer-review conferences (some CORE-indexed)

Conclusions (III): Published Papers (1)

Total numbers

- Two papers on JCR-indexed journals (SMC, JRR)
- Seven papers on international peer-review conferences (some CORE-indexed)
- Some more journals in progress. . .

Conclusions (III): Published Papers (1)

Total numbers

- Two papers on JCR-indexed journals (SMC, JRR)
- Seven papers on international peer-review conferences (some CORE-indexed)
- Some more journals in progress. . .

Security and Fault-Tolerant Techniques Modelling

- Rodríguez, R. J., Trubiani, C., and Merseguer, J. **Fault-Tolerant Techniques and Security Mechanisms for Model-based Performance Prediction of Critical Systems**. In *Proceedings of the 3rd International Symposium on Architecting Critical Systems (ISARCS)*, pages 21–30. ACM.
- Rodríguez, R. J., Merseguer, J. and Bernardi, S. **Modelling and Analysing Resilience as a Security Issue within UML**. In *Proceedings of the 2nd International Workshop on Software Engineering for Resilient Systems (SERENE)*, ACM, 2010, 42–51.
- Rodríguez, R. J. and Merseguer, J. Giese, **Integrating Fault-Tolerant Techniques into the Design of Critical Systems**. In *Proceedings of the 1st International Symposium on Architecting Critical Systems (ISARCS)*, Springer, 2010, 6150, 33–51.

Conclusions (III): Published Papers (2)

Performance Analysis and Resource Optimisation

- Rodríguez, R. J., Júlvez, J., and Merseguer, J. **On the Performance Estimation and Resource Optimisation in Process Petri Nets**. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP(99):1–14. doi: 10.1109/TSMC.2013.2245118
- Rodríguez, R. J., Júlvez, J., and Merseguer, J. **Quantification and Compensation of the Impact of Faults in System Throughput**. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. Accepted for publication.
- Rodríguez, R. J., Júlvez, J., and Merseguer, J. **PeabraiN: A PIPE Extension for Performance Estimation and Resource Optimisation**. In *Proceedings of the 12th International Conference on Application of Concurrency to System Designs (ACSD)*, pages 142–147. IEEE.
- Rodríguez, R. J. and Júlvez, J. **Accurate Performance Estimation for Stochastic Marked Graphs by Bottleneck Regrowing**. In *Proceedings of the 7th European Performance Engineering Workshop (EPEW)*, Springer, 2010, 6342, 175–190.

Conclusions (III): Published Papers (2)

Performance Analysis and Resource Optimisation

- Rodríguez, R. J., Júlvez, J., and Merseguer, J. **On the Performance Estimation and Resource Optimisation in Process Petri Nets**. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP(99):1–14. doi: 10.1109/TSMC.2013.2245118
- Rodríguez, R. J., Júlvez, J., and Merseguer, J. **Quantification and Compensation of the Impact of Faults in System Throughput**. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. Accepted for publication.
- Rodríguez, R. J., Júlvez, J., and Merseguer, J. **PeabraiN: A PIPE Extension for Performance Estimation and Resource Optimisation**. In *Proceedings of the 12th International Conference on Application of Concurrency to System Designs (ACSD)*, pages 142–147. IEEE.
- Rodríguez, R. J. and Júlvez, J. **Accurate Performance Estimation for Stochastic Marked Graphs by Bottleneck Regrowing**. In *Proceedings of the 7th European Performance Engineering Workshop (EPEW)*, Springer, 2010, 6342, 175–190.

Performance Analysis Applied to Scientific Workflows

- Rodríguez, R. J., Tolosana-Calasanz, R., and Rana, O. F. **Dynamic Data-Throttling in Scientific Workflows**. To be submitted to *IEEE Transactions in Parallel and Distributed Computing*.
- Rodríguez, R. J., Tolosana-Calasanz, R., and Rana, O. F. **Automating Data-Throttling Analysis for Data-Intensive Workflows**. In *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 310–317. IEEE.
- Rodríguez, R. J., Tolosana-Calasanz, R. and Rana, O. F. **Measuring the Effectiveness of Thottled Data Transfers on Data-Intensive Workflows**. In *Proceedings of the 6th International KES Conference on Agents and Multi-agent Systems – Technologies and Applications*, Springer, 2012, 7327, 144–153.

Acknowledgments

Who support me (financially)

- Grupo de Investigación en Sistemas de Eventos Discretos
- DISC (INFSO-ICT-224498)
- DPI2006-15390, DPI2010-20413
- TIN2007-66523
- DGA (CONAID) and CAI, ref. n. IT 5/11
- Ministry of Education, Culture and Sport, ref n. TME2011-00255

Acknowledgments

Who support me (financially)

- Grupo de Investigación en Sistemas de Eventos Discretos
- DISC (INFSO-ICT-224498)
- DPI2006-15390, DPI2010-20413
- TIN2007-66523
- DGA (CONAID) and CAI, ref. n. IT 5/11
- Ministry of Education, Culture and Sport, ref n. TME2011-00255

Who also support me (personally)

- Jorge & Merse
- DIIS staff
- Family, friends
- *

Performance Analysis and Resource Optimisation of Critical Systems Modelled by Petri Nets

Ricardo J. Rodríguez

Ph.D. DISSERTATION

Advisors: Dr. **Jorge Emilio Júlvez Bueno** and Dr. **José Javier Merseguer Hernáiz**

Dpto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza

June 24th, 2013

Zaragoza, Spain