# Grid-enabled Performance Analysis using Stochastic Logics

## The GRAIL Project

William Knottenbelt
Tamas Suto

Department of Computing
Imperial College London

17 November 2005, SENVA'05

# The AESOP Research Group

- **A**nalysis, **E**ngineering, **S**imulation & **O**ptimisation of **P**erformance
- Quantitative evaluation of systems
- Stochastic Petri Nets, Stochastic Process Algebras, Queueing Networks
- Stochastic logics and specification of performability requirements
- Parallel and distributed solution techniques for Markov and semi-Markov chains

# Available Performance Tools

- *DNAmaca*
  - Markov chain steady state analyser
  - Analyses models with state spaces larger than $O(10^7)$
- *SMCA*
  - Semi-Markov chain steady state analyser
  - Analyses models with state spaces larger than $O(10^7)$
- *HYDRA*
  - Passage time and transient analyser for very large Markov models
  - Uses parallel hypergraph partitioning and uniformisation
- *SMARTA*
  - Semi-Markov passage time analyser
  - Implements an iterative passage time algorithm
  - Uses parallel hypergraph partitioning

# Grid Facilities at Imperial College London

- Nebula cluster (32 proc.)
  - 32 proc., 64 core 270 Opteron, 4GB RAM, Infinipath HTX
- Luna cluster (16 proc.)
  - 8 dual 3GHz P4 Xeon, 2GB RAM, Gigabit Ethernet
- Mars cluster (404 proc.)
  - 72 dual 1.8GHz Opteron, 2GB RAM, Infiniband
  - 40 dual 1.8GHz Opteron, 4GB RAM, Gigabit Ethernet
  - 88 dual 1.8GHz Opteron, 2GB RAM, Gigabit Ethernet
  - 4 dual 2.2GHz Opteron, 4GB RAM, Gigabit Ethernet
- Viking cluster (512 proc.)
  - 96 dual 2GHz P4 Xeon, 2GB RAM, Myrinet / Fast Ethernet
  - 32 dual 2GHz P4 Xeon, 1GB RAM, Fast Ethernet
  - 128 dual 2.8GHz P4, 2GB RAM, Myrinet / Fast Ethernet

# Research Objectives

- Develop new theoretical methods in stochastic logics and model-checking
- Enable the formal verification of natural language-based performance requirements in industrial-scale models
- Use distributed and parallel computing in a Grid environment for large-scale model checking

# Applications

- Health Care
  - Need response time analysis of patient flow models to achieve QoS improvement amidst ever growing service demand
- Telecommunications Industry
  - Need response time guarantees for wireless services
  - QoS modelling and analysis for strategic planning essential
- Financial Sector
  - Need models and tools for predicting run-time performance of critical applications

# Continuous Stochastic Logics

- Enable the rigorous, verifiable, expressive and composable specification of complex performance requirements using logical formulae
- Various different flavours: CSL, CSRL, aCSL, eCSL, *etc.*
- BUT: verification of logical formulae can be computationally very intensive on larger models

# Extended Continuous Stochastic Logic (eCSL)

- State-based logic adapted to work on the SMSPN model level
- Can express a richer class of passage time quantities than CSL
- Can represent requirements based on transient and steady-state distributions
- Simple formulae can be composed into compound queries that are verified on a model

# The Power of eCSL

- Can express the availability, reliability and response-time requirements of a semi-Markovian system in a single logical formula

$$\vec{m} \models \underbrace{\mathcal{S}_{P_1}(\psi_1)}_{\text{availability}} \wedge \underbrace{\mathcal{T}_{P_2}^{T_2}(\psi_2, \psi_3)}_{\text{reliability}} \wedge \underbrace{\mathcal{P}_{P_3}^{T_3}(\psi_4, \psi_5)}_{\text{response time}}$$

# An Example of an eCSL Formula

- "Does a passage occur within 10 seconds with at least 90% probability?"

$$\mathrm{Sat}(p_1[35] \wedge p_5[10]) \models \mathcal{P}^{[0,10)}_{\{0.9,1\}}(p_2[175], p_6[1])$$

# An Example of an eCSL Formula

$$\mathrm{Sat}(p_1[35] \wedge p_5[10]) \models \mathcal{P}^{[0,10)}_{\{0.9,1\}}(p_2[175], p_6[1])$$

- Here, the passage is defined by:
  1. The satisfiability expression on the left → start states of the passage
  2. The first argument of the $\mathcal{P}$ tuple → target states of the passage
  3. The second argument of the $\mathcal{P}$ tuple → excluded states through which the passage must not pass
- The $p_n[m]$ expressions define sets of states from the Petri net model

# Problem 1

Problem: Stochastic logics are too abstract and difficult to understand

Solution: Specify complex QoS requirements in natural language and with graphical means $\rightarrow$ map automatically onto eCSL (or similar)

- Simplifies requirements specification
- Maintains expressiveness and analysis power
- No understanding of logical formalism required

# Problem 2

Problem: Presently, the mapping of eCSL formulae onto interface languages of performance analysis tools has to be performed manually

Solution: Automatic query mapping

# Problem 3

Problem: Limited solution capacity → no industrial-scale models

Solution: Integrate Grid cluster as computational backbone → vastly extended solution capacity

# Problem 4

Problem: No automated method for decomposing formulae and scheduling the execution of sub-computations

Solution: Automatic analysis of execution dependencies to produce optimised distribution of computation

# Outlook

- Existing performance tools will be enhanced for interfacing with a Grid back-end
- eCSL formulae will be dissected and resulting sub-computations delegated to respective tools
- Tools will perform parallelised model checking using the Grid back-end's extensive computing power

# Accessible Front-End

- We have the tools, but lack a coordinating front-end
- We want to make the power of the underlying stochastic logics accessible to system designers through the front-end

# Performance Requirement Specification

1. Guided natural language-based performance requirement specification, aided by graphical methods
2. Automatic translation to eCSL
3. Visualisation as an interactive Query Tree to hide the complexity of eCSL and enhance accessibility

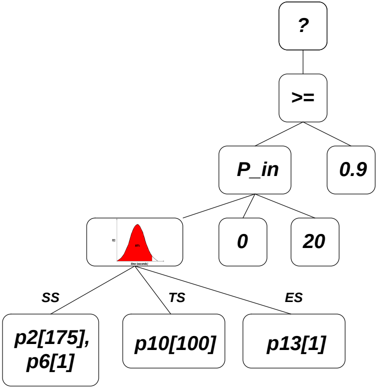
# An Example

- "Does a passage between the set of states $p_2[175] \wedge p_6[1]$ and $p_{10}[100]$ occur within the time interval [0,20] with at least 90% probability, given that the set of states $p_{13}[1]$ must not be reached along the passage?"
- In eCSL:

$$\mathrm{Sat}(p_2[175] \wedge p_6[1]) \models \mathcal{P}^{[0,20)}_{\{0.9,1\}}(p_{10}[100], p_{13}[1])$$

# An Example

- Translated to Query Tree representation:



# Extensions to eCSL

- Allow the representation of an even wider range of performance requirements
- Arsenal of possible model-level questions will be enriched, incorporating reward structures and action-based constraints

# Query Processing

- Queries will be dissected and analysed for evaluation optimisation opportunities
- Optimised Grid scheduling strategies will be developed

# Conclusions

- Objective:
  - To realise industrial-scale model checking, using stochastic logics and a Grid-based computational infrastructure, controlled by an accessible user interface
- Vision:
  - To integrate our efforts with other projects to realise a Europe-wide model checking Grid

Thank you for your attention.

Any questions ?

{*wjk,suto*}@*doc.ic.ac.uk*