Grid-enabled Performance Analysis using Stochastic Logics

Tamas Suto

Department of Computing
Imperial College London

8 September 2005

PASTA'05

Aim of Research

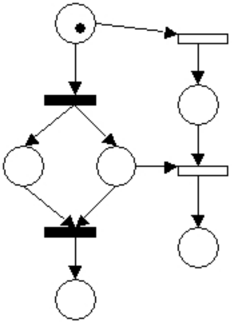
- Develop new theoretical methods in stochastic logics and model-checking
- To enable the formal verification of natural language-based performance requirements in industrial-scale models
- Using distributed and parallel computing in a Grid environment

Relevance

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

Stochastic Petri Nets

- A useful high-level modelling formalism for representing complex real-life systems
- Underlying Markov chains can be analysed for various performance measures of interest



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
- BUT: verification of logical formulae can be computationally very intensive on larger models

Extended Continuous Stochastic Logic (eCSL)

- One of the latest variants adapted to work on the SMSPN model level
- Allows to express a richer class of passage time quantities than CSL, as well as to specify requirements based on transient and steady-state distributions
- Simple formulae can be composed to form compound queries that can be verified against a model

The Power of eCSL

- Its ability to express, in a single compound logical formula, the availability, reliability and response-time requirements of a semi-Markovian system

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

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 hard to understand

Solution: Specify complex QoS requirements in natural language, together with graphical methods → map them automatically onto eCSL

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

Problem 2

Problem: Mapping of eCSL formulae onto interface languages of existing analysis tools has to be performed manually

Solution: Automatic performance query mapping

Problem 3

Problem: Limited solution capacity → no industrial-scale models

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

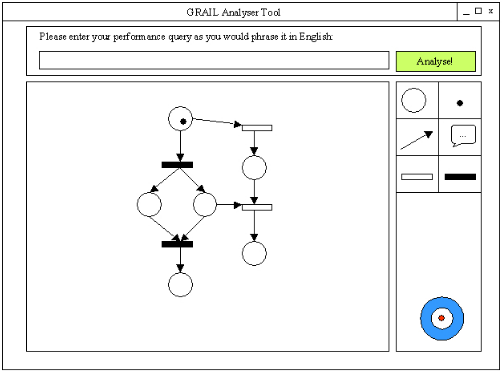
Problem 4

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

Performance Requirements Specification

- Natural language input processing aided by graphical specification methods

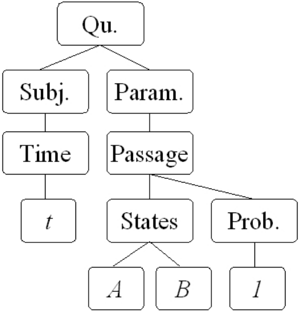


Performance Requirements Mapping

- eCSL will serve as the intermediary between specification and computation
- Hence, performance requirements need to be mapped onto eCSL
- The eCSL formulae will then be decomposed into a semantic tree-style representation that facilitates optimisation of computation

An Example

- “What is the time required to perform a passage between the sets of states A and B ?”



Extensions to eCSL

- Expressiveness of eCSL will be further extended to allow the representation of an even wider range of performance requirements
- Arsenal of possible model-level questions will be enlarged, incorporating reward structures and transition-specific metrics

Tool Development

- A tool featuring all achievements of our research will be developed
- With applications in a wide range of academic and industrial fields

Query Processing

- Establish performance requirements and represent in a logical formula
- Disseminate and analyse for opportunities of evaluation optimisation

Distribution and Grid Scheduling

- Query execution will be distributed amongst existing analysis tools
- Tools will become Grid-enabled and computation will be distributed and parallelised on a Grid architecture
- Optimal Grid scheduling strategies will be developed to maximise the Grid technology exploitation

Thank you for your attention.

Any questions?