GRAIL : Grid-enabled Performance Analysis using Stochastic Logics

Tamas Suto

Department of Computing, Imperial College London,
South Kensington Campus, London SW7 2AZ, UK
suto@doc.ic.ac.uk

Abstract. This paper aims at introducing our latest research efforts in the fields of performance analysis, stochastic logics and Grid technology adaptation. We are currently exploring a new approach for a human-friendly and intelligent graphical specification of Quality of Service-related performance requirements for industrial-scale real-life systems. Such systems are modelled by stochastic Petri nets, on which the necessary analyses are performed. To ensure formality, expressiveness and to allow for a concise but yet unambiguous representation of potentially very complex and compound performance requirements, a new variant of continuous stochastic logics will be employed. Resulting logical formulae will be disseminated and analysed for an optimal distribution on a computational Grid back-end for the purpose of model-checking previously untractable large-scale models. The research will eventually result in a new form of stochastic logic that will feature enhanced performance requirement representation capabilities, thereby extending the depth and breadth of possible performance measurement scenarios, as well as in the development of the means to utilise for the first time the computational power of the Grid for performance analysis by distributing, parallelising and optimising required model-checking computation on a Grid cluster. In this way, available solution space will be vastly extended and truly industrial-scale systems finally become analysable.

1 Introduction

The power to reason about response times is an essential ingredient in providing Quality of Service (QoS) guarantees in almost all concurrent and distributed systems, including mobile phone networks, Web and database servers, embedded systems, stock market trading platforms and health care systems[1]. Together with transient and steady-state measures, response time analysis provides system designers and engineers with a comprehensive arsenal of performance tools.

To date, specification of key QoS performance requirements has been on an informal and restricted basis. It has not been possible, for example, to specify *complex* constraints, such as restrictions on percentiles of waiting times, nor to specify *compound* queries whereby several QoS requirements must be met simultaneously. A promising way forward has presented itself with the emergence of temporal stochastic logics. In particular, continuous stochastic logics (CSLs), which enable the rigorous, verifiable, expressive and composable specification of complex performance requirements using logical formulae. However, the verification of these formulae can be computationally very intensive on larger models. Our research therefore aims to combine the expressive power of an extended CSL with a novel approach to formula evaluation using the Grid, which will enable the analysis of considerably larger models than presently possible.

Temporal logics have different flavours, among them pCTL [Han94] for discrete-time probabilistic verification, CSL [Azi96,Azi00] applied to Markovian state spaces [Bai99,Bai00,Kat01,Bai03], aCSL [Her00b] applied to process algebras and CSL/eCSL [You02,Inf01,Bra03] for semi-Markov models. eCSL augments CSL with the ability to express a richer class of passage time quantities, as well as the ability to specify requirements based on transient and steady-state distributions. Unlike basic CSL, which operates at the state-transition level, eCSL operates at the model level.

Temporal logics are popular in some academic circles [Kwi02,d'A04,Her00a]; however, their use in industry is limited. We have identified three main reasons for this:

[1] such as the requirement of Accident & Emergency units in the UK to see, treat and discharge 90% of patients in under four hours.

1. Stochastic logics are too "esoteric" for most system designers, in that it is challenging for non-specialists to understand and formulate logical queries.
2. The mapping of stochastic logic formulae onto the interface languages of existing modelling and analysis tools has to be performed manually, requiring specialist knowledge. Furthermore, there is no automated method for decomposing formulae and scheduling the execution of resulting sub-computations.
3. The limited solution capacity of most existing tools incorporating stochastic logics hinders the evaluation of unrestricted industrial-size models.

2 Research Objectives

Our research will aim to address these issues in the following ways:

1. Issue (1) is addressed by investigating new ways to specify, at the model level, using natural language and graphical input mechanisms, complex QoS requirements, which can be mapped automatically onto stochastic logic formulae. This will drastically simplify the requirements specification process, while maintaining expressiveness and analysis power. No understanding of the underlying logical framework will be necessary on the end-user's side.
2. Issue (2) is addressed by developing novel techniques for the automatic mapping of stochastic logic formulae onto the interface languages of performance analysis tools developed during the PASTRAMI project[2], and by the automatic analysis of execution dependencies to produce optimised Grid execution schedules.
3. Issue (3) is addressed by making use of the vast amount of computing power that the Grid provides. Small-cluster computation is not sufficient for the evaluation of industrial-size models; therefore the resources of an integrated Grid cluster will serve as the computational backbone for our approach.

The primary goal of our research will be the development of new theoretical methods in stochastic logics and Grid-enabled software tools that implement these techniques. It will introduce a stochastic model-checking method that for the first time will realise the seamless formal verification of graphically-specified model-level QoS performance requirements in industrial-scale models.

3 Applicability

This research will have significant practical applications. In particular, immediate benefits will arise in the following areas:

- Response times are of crucial importance throughout numerous disciplines of engineering. Telecommunications providers, for example, require response time guarantees when delivering wireless services. Strategic planning in telecoms companies involves large-scale QoS modelling and analysis.
- Public health care institutions require response time analysis of patient flow models to help improve patient-perceived QoS (as measured by the 90^{th} waiting time percentile) amidst ever growing service demand.
- Companies in the financial and other sectors are increasingly investing in Grid-related technologies that allow for the virtualisation of both computation and storage. The key issue in using Grids effectively is to ensure predictability and deliver on quantitative and stochastic SLA's for general applications. As the code is migrated to unknown machines the only way to achieve predictable and efficient performance is through techniques such as performance models. Stochastic logics are ideal in this context.

Academic research will also benefit from our work. Close inter-departmental collaboration will be established at Imperial College London to aid ongoing research efforts in other disciplines of science, such as Biology[3]

[2] http://aesop.doc.ic.ac.uk/projects/pastrami
[3] Relevant projects include *eProtein*, aiming at providing a structure-based annotation of the proteins in the major genomes, and *eSupport*, aiming at automatic high-throughput annotation of genomes.

and Physics[4]. The research and development will also complement other currently ongoing projects in the Department of Computing[5].

4 Theoretical Background

4.1 Extended Continuous Stochastic Logic

Formal logics for stochastic systems provide a concise and rigorous way to pose performance questions and allow for the composition of simple queries into more complex ones. One such logic is CSL, which can express performance measures by selecting states and paths from a system that meet steady-state and passage time criteria.

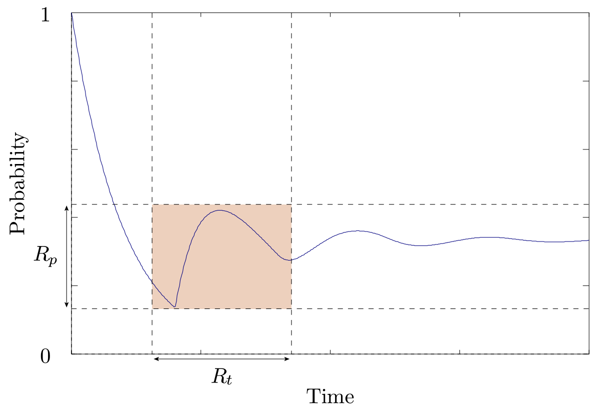

Fig. 1. A graphical representation of a logical constraint on system reliability, using a $\mathcal{T}$-formula

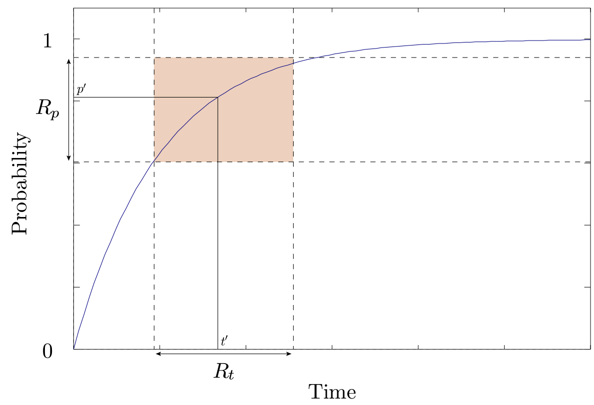

Fig. 2. A graphical representation of a logical constraint on response-time, using a $\mathcal{P}$-formula

Further development of this logic has resulted in eCSL, an extended version that operates on semi-Markov stochastic Petri nets (SMSPNs), from which underlying semi-Markov processes can be generated. The power of eCSL lies in its ability to express, in a single compound logical formula, the reliability, availability and response-time requirements of a semi-Markovian system.

The syntax of an eCSL statement Ψ defined over an SMSPN with set of markings M is given by:

$$\Psi \stackrel{\text{def}}{=} \mathsf{tt} \mid \Psi \wedge \Psi \mid \neg\Psi \mid \mathcal{S}_{\boldsymbol{\rho}}(\psi) \mid \mathcal{T}_{\boldsymbol{\rho}}^{\boldsymbol{\tau}}(\psi,\psi) \mid \mathcal{P}_{\boldsymbol{\rho}}^{\boldsymbol{\tau}}(\psi,\psi)$$

$$\psi \stackrel{\text{def}}{=} \mathsf{tt} \mid \psi \wedge \psi \mid \neg\psi \mid p[N]$$

Figure 1 shows an example of a reliability constraint $\boldsymbol{m} \models \mathcal{T}_{R_p}^{R_t}(\psi_1,\psi_2)$ which reasons about the transient distribution in the shaded area. It expresses the requirement that the probability of system failure should lie within the region R_p over the time region R_t. If any part of the transient function over R_t lies outside the region R_p, the property fails.

Figure 2 shows an example response time constraint $\boldsymbol{m} \models \mathcal{P}_{R_p}^{R_t}(\psi_1,\psi_2)$ which reasons about a passage time distribution. It expresses the requirement that there should exist a response time, $t' \in R_t$, with (cumulative)

[4] Such as *GridPP*, a project aimed at developing a Grid for particle physics.
[5] Such as *PASTRAMI*, focusing on passage time analysis in large Markov and semi-Markov chains, and *ARTERY*, a project on response time analysis in health care systems.

probability of occurrence, $p' \in R_p$. Note that this is applied to the cumulative distribution (CDF) function of a passage time; thus if the shaded region does not intersect the CDF, the property fails.

As an example of how eCSL is used to pose a performance question, consider the following:

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

This formula expresses the question: "Does a passage time occur within 10 seconds with at least 90% probability?" In this case, the passage is defined by:

(i) the satisfiability expression on the left, which represents the start states of the passage;
(ii) the first argument of the $\mathcal{P}$ tuple, which represents the target states of the passage;
(iii) the second argument of the $\mathcal{P}$ tuple, which represents the excluded states through which the passage must not pass

The $p_n[m]$ expressions define sets of states from the Petri net model. For instance $p_1[35] \wedge p_5[10]$ selects all the Petri net markings (in M) which have 35 tokens on place p_1 and 10 tokens on place p_5.

Simple formulae can be composed to form compound queries which can be verified against a model. For example, the generic formula below represents the logical requirement that simultaneous availability, reliability and response time properties must hold:

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

eCSL model-checking is a complex and computationally intensive process, particularly when models are large. Simple parallelisation itself is not a good solution to this problem, since the computing power required for checking industrial-size models is not available in traditional clusters. Structural and dependency analysis of eCSL queries will allow us to perform query decomposition, optimise parallelisation and avoid duplicate computations, through the use of speculative caching and other techniques.

Examples of Grid-style optimisations which derive directly from the compound logical nature of eCSL might include:

Overlapping steady-state calculations $\mathcal{S}_{P_1}(\psi_1) \wedge \mathcal{S}_{P_2}(\psi_2)$. If there is an overlap in the states satisfying ψ_1 and ψ_2, then it may be that the overall conjunction is satisfied by the accumulated steady-state on the intersection alone. So if $\mathcal{S}_{P_1 \cap P_2}(\psi_1 \wedge \psi_2)$ is satisfied by the model we would not have to perform the full calculation $\mathcal{S}_{P_1}(\psi_1) \wedge \mathcal{S}_{P_2}(\psi_2)$; this sub-calculation could be computed first by an eCSL-aware scheduler.

Contained response time calculations $\mathcal{P}^{T_1}_{P_1}(\psi_1, \psi) \wedge \mathcal{P}^{T_2}_{P_2}(\psi_2, \psi)$. For response time calculations which share the same excluded state specification (ψ) and which intersect in time/probability space, a potential optimisation can be used if target states of one passage definition are contained in the target states of the other, that is, $\mathrm{Sat}(\psi_2) \subseteq \mathrm{Sat}(\psi_1)$. Thus full validation of $\mathcal{P}^{T_1}_{P_1}(\psi_1, \psi) \wedge \mathcal{P}^{T_2}_{P_2}(\psi_2, \psi)$ can be avoided if $(\psi_1 \Rightarrow \psi_2) \wedge \mathcal{P}^{T_1}_{P_2}(\psi_2, \psi)$ and $P_2 \subseteq P_1$, $T_2 \subseteq T_1$.

Our aim is to exploit these techniques in the context of the Grid to enable efficient stochastic model-checking of industrial-scale models.

5 Outlook

We are about to enter the development stage, which will in the end contribute to the community a useful and widely applicable software tool that will provide advanced means for carrying out performance analyses on a much wider range of models, exploiting state-of-the-art methodologies and infrastructure.

Our efforts will also contribute to performance analysis research with various novel techniques for model-checking, efficient scheduling and distribution methods for Grid integration and possibly new or enhanced logical formalisms for a more flexible representation of performance requirements.

References

[Azi96] A. Aziz, K. Sanwal, V. Singhal, R. Brayton. *Verifying Continuous-Time Markov Chains.* In *Computer-Aided Verification*, vol. 1102 of *Lecture Notes in Computer Science*, pp. 269–276. Springer-Verlag, 1996.

[Azi00] A. Aziz, K. Sanwal, V. Singhal, R. Brayton. *Model Checking Continuous-Time Markov Chains.* ACM Transactions on Computational Logic, 1(1), pp. 162–170, 2000.

[Bai99] C. Baier, J.-P. Katoen, H. Hermanns. *Approximate Symbolic Model Checking of Continuous-Time Markov Chains.* In *Proc. 10th Intl. Conf. on Concurrency Theory (CONCUR'99)*, vol. 1664 of *Lecture Notes in Computer Science*, pp. 146–162. Springer-Verlag, 1999.

[Bai00] C. Baier, B. Haverkort, H. Hermanns, J.-P. Katoen. *Model Checking Continuous-Time Markov Chains by Transient Analysis.* In *Proc. 12th Intl. Conf. on Computer Aided Verification (CAV'00)*, vol. 1855 of *Lecture Notes in Computer Science*, pp. 358–372. Springer-Verlag, Chicago, July 2000.

[Bai03] C. Baier, B. R. Haverkort, H. Hermannsr, J.-P. Katoen. *Model Checking Algorithms for Continuous-Time Markov Chains.* IEEE Transactions on Software Engineering, 29(6), pp. 524–541, July 2003.

[Bra03] J. T. Bradley, N. J. Dingle, W. J. Knottenbelt, P. G. Harrison. *Performance Queries on Semi-Markov Stochastic Petri Nets with an Extended Continuous Stochastic Logic.* In *Proc. Petri Nets and Performance Models (PNPM'03)*, pp. 62–71. IEEE Computer Society, September 2003.

[d'A04] D. d'Aprile, S. Donatelli, J. Sproston. *CSL Model Checking for the GreatSPN Tool.* In *Proc. 19th Int. Symposium on Computer and Information Sciences (ISCIS 2004)*, vol. 3280 of *Lecture Notes In Computer Science*, pp. 543–552. Springer-Verlag, 2004.

[de 01] L. de Alfaro, S. Gilmore (eds.). *Proc. Process Algebra and Probabilistic Methods (PAPM-PROBMIV 2001)*, vol. 2165 of *Lecture Notes in Computer Science*. Springer-Verlag, Aachen, Germany, September 2001.

[Han94] H. Hansson, B. Jonsson. *A Logic for Reasoning about Time and Reliability.* Formal Aspects of Computing, 6(5), pp. 512–535, 1994.

[Her00a] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, M. Siegle. *A Markov Chain Model Checker.* In *Proc. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'00)*, pp. 347–362, 2000.

[Her00b] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, M. Siegle. *Towards Model Checking Stochastic Process Algebra.* In *Proc. 2nd Int. Conf. on Integrated Formal Methods (IFM'00)*, pp. 420–439, November 2000.

[Inf01] G. Infante López, H. Hermanns, J.-P. Katoen. *Beyond Memoryless Distributions: Model Checking semi-Markov Chains.* In de Alfaro and Gilmore [de 01], pp. 57–70.

[Kat01] J.-P. Katoen, M. Kwiatkowska, G. Norman, D. Parker. *Faster and Symbolic CTMC Model Checking.* In de Alfaro and Gilmore [de 01], pp. 23–38.

[Kwi02] M. Kwiatkowska, G. Norman, D. Parker. *Probabilistic Symbolic Model Checking with PRISM: A Hybrid Approach.* In *Proc. Tools and Algorithms for Construction and Analysis of Systems (TACAS'02)*, vol. 2280 of *Lecture Notes in Computer Science*, pp. 52–66. Springer-Verlag, Grenoble, France, April 2002.

[You02] H. L. S. Younes, R. G. Simmons. *Probabilistic Verification of Discrete Event Systems using Acceptance Sampling.* In *Proc. 14th Intl. Conference on Computer Aided Verification (CAV 2002)*, vol. 2404 of *Lecture Notes in Computer Science*, pp. 223–235. Springer-Verlag, July 2002.