Invited Response to Computer Journal Lecture by Prof. Jane Hillston

Jeremy T. Bradley[1], Nigel Thomas[2],
Richard A. Hayden[1] and Anton Stefanek[1]

[1]*Department of Computing, Imperial College London, UK*
[2]*School of Computing Science, Newcastle University, UK*
Email: jb@doc.ic.ac.uk, nigel.thomas@ncl.ac.uk, {as1005,rh}@doc.ic.ac.uk

Keywords: stochastic process algebra, PEPA, mean field analysis, fluid analysis, scalability, performance modelling

Received October 2011; revised October 2011

1. INTRODUCTION

The introduction of stochastic process algebra (SPA) has had a profound impact on the field of performance modelling. Hillston's PEPA has been at the forefront of this development [1]. There are a number of reasons why the use of stochastic process algebra is attractive to the stochastic modeller. The parsimonious set of operators creates an almost programming-like simplicity to model specification, meaning complex behaviours can be modelled in a concise and relatively understandable way. The models, although complex, can be analysed to show that they are deadlock free and that intended behaviours are reachable in its evolution (unlike simulation). The formal underpinning of the algebra means that models can be derived from other formal (or semi-formal) specifications in an automatic or semi-automatic way. This formality also means that the process algebra model can itself be manipulated into provably equivalent alternative forms that are more readily solved by numerical analysis. In the case of PEPA, the specification and analysis is also supported by a powerful set of modelling tools [2, 3, 4, 5, 6].

Stochastic process algebra clearly provides a very convenient means for specifying large, detailed stochastic models. Their inherently compositional nature means that models can be constructed piece by piece and then combined to form the overall behaviour, much as one would construct an actual system. The main problem with this approach is that it becomes all too easy to specify a model beyond the bounds of traditional analysis. This is because SPA models suffer from the well known problem of *state space explosion*, where each additional component causes a multiplicative increase in the size of the global state space. Thus analysis based on studying the underlying continuous time Markov chain (CTMC), becomes prohibitively expensive to perform. This problem is particularly significant when there are many instances of the same type of component (so-called *massively parallel systems*). Such models may be extremely concise to specify, but even when the state space is folded or lumped [7], it may still far exceed the capacity available for solution.

There have been many attempts to find efficient solutions to large stochastic process algebra (SPA) models. Many of these approaches have been based on concepts of decomposition, such as product form solutions [8, 9]. Applying such approaches to stochastic process algebra allows the concepts to be understood in a more general modelling framework and applied to non-queueing models. This form of analysis still relies on studying the CTMC, but the entire state space does not need to be used in computation, thus avoiding the problems associated with manipulating very large matrices, which is generally the limiting factor in CTMC analysis. Symbolic techniques involving MTBDD data structures can be used to compress the storage of the underlying CTMC as generated from a stochastic process algebra model [10, 11]. An alternative approach is to avoid deriving the CTMC directly, for instance by Kronecker representation [12] or mean value analysis [13]. Such approaches are efficient in memory (as the CTMC is never stored in its entirety), but can be restrictive in terms of the solution time or the metrics of interest that can be derived.

Hillston [14] has taken a very different approach, inspired by systems biology [15, 16], by deriving a fluid approximation based on ordinary differential equations

(ODEs). This approximation maps a stochastic model specification on to a deterministic representation, thus the intrinsic randomness of the system is lost. However, the approach is extremely scalable and there are instances where the approximate solution is extremely accurate [5]. This is particularly the case when the model is very large, which coincides with the situation where state space based methods fail. In addition the ODE solution provides an insight into the transient behaviour of the system, which is generally even more costly to compute by direct CTMC analysis.

2. FLUID ANALYSIS AND SCALABILITY

Since Hillston's key paper on fluid modelling in PEPA [14], there has been a substantial interest in the *fluid* or *mean-field* approach to analysing stochastic problems in computing [17, 18, 19, 20, 21, 22] and stochastic process algebra models in particular [23, 24, 25, 26, 27].

Fluid analysis approximates the mean [14, 17] and higher-moment dynamics [28, 25] of a complex stochastic process using ordinary differential equations. The major benefit that fluid analysis brings is the ability to produce quantitative analysis of a large and complex stochastic system with very little computational effort. This in turn allows the modeller to trial many different system configurations and parameterisations easily whereas in the past a single system instance would have taken many hours or days to analyse.

Latterly, fluid techniques have been used to extract useful passage time measures [29, 30] from systems specified using stochastic process algebra. This gives the ability to extract how long key transactions in a system will take. With the rapid computation of fluid analysis, we can now parameter sweep efficiently to find which model variables have greatest effect on a key passage time measure. Often these passage times will take the form of requirements or service level objectives (SLOs) for the system, for example, 97.8% of search queries should be responded to within 0.5 seconds. Understanding how a key passage time is sensitive to changes in model parameters can help improve the design of system. In particular how a passage time reacts to changes in the scale of component deployment within the system, so-called scalability analysis, is critical to the system engineering process.

2.1. Example: A Scheduled Client–Server

To illustrate the efficacy of fluid analysis for scalability testing, we set up a scheduled client–server model. Figure 1 represents a system with n clients and m servers and a scheduling component which allocates requests from clients to individual servers (from [31, Sect. 4]). We construct this model in a flavour of PEPA, called GPEPA [32, 25]. We then impose a service level objective for the system, that a client must complete its server access within t seconds with probability 0.95.

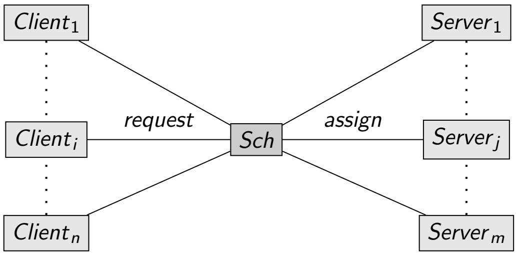

FIGURE 1. A scheduled client–server architecture

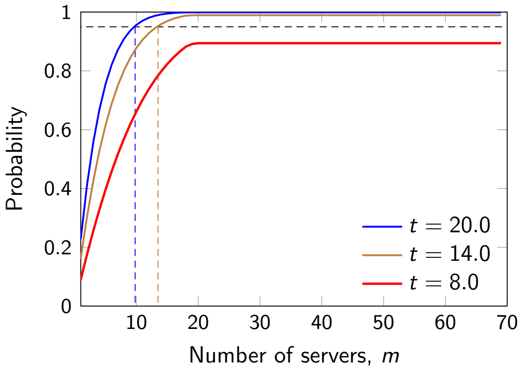

FIGURE 2. Varying the number of servers for distinct time thresholds, t, in the service level objective for $t = 20, 14$ and 8.

Figure 2 shows the probability that the service requirement is met for a certain time threshold t, as the number of servers m is varied from 0 to 70. We observe that for a service threshold of $t = 20$ seconds, we require about 10 servers to attain the required 0.95 probability. As we tighten the service threshold to $t = 14$ seconds, about 14 servers are needed to meet the SLO. However for a service threshold of $t = 8$ seconds, we see that for this particular model, no amount of servers will guarantee a 0.95 probability of meeting the service requirement. This is a nice indication that simply throwing resources at a problem will not necessarily solve it. In this case, we would have to redesign the servers, making them, for instance, individually faster or more responsive to the client requests in order to meet the requirement for $t = 8$ seconds.

2.2. Example: Energy Optimisation

In Markov reward models, accumulated rewards have been used to capture energy consumption in systems [33, 34]. The premise of reward models is to have a system-wide reward that is accumulated over time as the system waits in designated states. Many reward variables representing different aspects of system operation can be simultaneously maintained e.g. cost and income as well as energy. However the computational cost of analysing traditional Markov reward models is prohibitive and certainly prevents parameter sweeping for any realistic size of system.

A useful development in the fluid analysis of stochastic process algebra models, is the ability to analyse SPA-specified Markov reward models and specifically accumulating reward variables using fluid analysis [35]. This gives the modeller the same advantage of rapid analysis for large reward systems as fluid analysis techniques do for large stochastic process algebra models. With this addition to the fluid analysis toolset, the modeller is in a position to look for configurations and parameterisations of an energy-consuming system that will minimise the amount of energy being used. Fluid approximations have been used to capture system energy problems previously [36] but there are distinct benefits to doing this within the context of a stochastic process algebra framework, as discussed in Section 1.

Bringing all of these developments together allows a designer to consider a performance–energy trade-off in an SPA model. Performance objectives can be specified by means of service level objectives that a system must comply with. In doing so, the designer is constraining the design-space of the system. Within that constrained feasible design-space, fluid techniques can be deployed to look for the minimum energy configuration. This can be posed as a classic constrained optimisation problem [37].

To demonstrate this process, we extend the original scheduled client–server model of Section 2.1 to incorporate high and low priority clients and two classes of server (fast and slow) [31]. Further we impose two distinct service level objectives for the design of the system. The first SLO for low priority clients (SLO_L), states that 80% of requests sent from these clients should be serviced within 8 seconds; the second, for high priority clients (SLO_H) states that 90% of requests sent from these clients should be serviced within 6.5 seconds.

Figure 3 displays the performance-energy design space for this problem. We vary the population of fast and slow servers in the system. The shaded region shows the design space that satisfies the individual SLOs with the intersection of these regions satisfying both. For the feasible design space (satisfying both SLOs) we plot the energy consumption from the fluid reward computation. We can see that for a certain system configuration of 40 slow servers and 32 fast servers we are able to both satisfy the performance requirements and minimise the energy usage in the system.

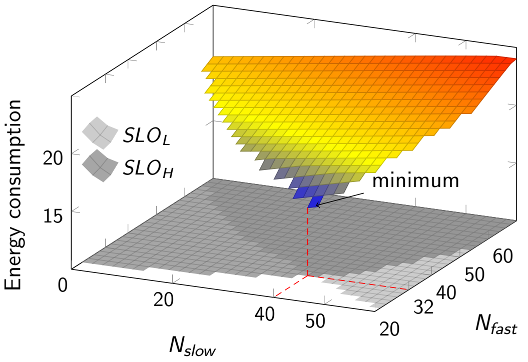

FIGURE 3. Energy minimisation over server configurations for given performance constraints

This example emphasises the power and scalability of Hillston's original fluid SPA analysis techniques. Many hundreds of system configurations can be tested via the solution of sets of ODEs with much reduced computational effort.

3. FINAL REMARKS

Hillston's original development of a novel analysis technique for stochastic process algebra models has made a substantial impact. In particular performance analysis computations that used to take many hours or days can be completed in seconds. It has been shown how this potential for rapid quantitative analysis can allow a system designer to explore many design possibilities with low computational effort. Extensions in fluid computation of SPA models can be shown to produce passage time and reward analysis which can allow systems to be engineered that consume less energy, while maintaining their performance criteria.

Many novel areas of application have been explored through the fluid flow approach. For example the study of Internet worm attacks [38], properties of security protocols [39], analysis of distributed learning environments [40] and crowd behaviour [41].

While this technique has been a tremendous advance, there are many challenges that remain to be tackled. The fluid approach works best on systems with many replicated components, therefore how can we take advantage of this analysis for systems which have more discrete on–off behaviour? With the advent of fluid analysis of rewards, we would like to introduce hysteresis between rewards and system behaviour; there is work to be done in establishing convergence properties for this type of control-based system. Can we incorporate more general timing information in our model and still have meaningful fluid analysis techniques. The next few years will hopefully see many new and exciting developments in these and other areas.

Acknowledgements

The authors are supported by an EPSRC funded project on the Analysis of Massively Parallel Stochastic Systems (EP/G011737/1 and EP/G011389/1).

REFERENCES

[1] Hillston, J. (1996) *A Compositional Approach to Performance Modelling*, Distinguished Dissertations in Computer Science, **12**. Cambridge University Press.

[2] Gilmore, S. and Hillston, J. (1994) The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. *Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Vienna, May, Lecture Notes in Computer Science, **794**, pp. 353–368. Springer-Verlag.

[3] Clark, G., Gilmore, S., Hillston, J., and Thomas, N. (1999) Experiences with the PEPA performance modelling tools. *IEE Proceedings – Software*, **146**, 11–19.

[4] Tribastone, M. (2007) The PEPA plug-in project. *QEST'07, Proceedings of the 4th Int. Conference on the Quantitative Evaluation of Systems*, September, pp. 53–54. IEEE Computer Society.

[5] Stefanek, A., Hayden, R. A., and Bradley, J. T. (2010) A new tool for the performance analysis of massively parallel computer systems. *QAPL'10, 8th Workshop on Quantitative Aspects of Programming Languages*, June, Electronic Proceedings of Theoretical Computer Science, **28**, pp. 159–181.

[6] Smith, M. and Gilmore, S. (2011) Visualisation for stochastic process algebras: The graphic truth. In Thomas, N. (ed.), *Proceedings of the 8th European Performance Engineering Workshop on Computer Performance Engineering (EPEW)*, October, Lecture Notes in Computer Science, **6977**, pp. 310–324. Springer-Verlag.

[7] Kemeny, J. G. and Snell, J. L. (1960) *Finite Markov Chains*. Van Nostrand.

[8] Hillston, J. (2001) Exploiting structure in solution: Decomposing composed models. *Lectures on Formal Methods and Performance Analysis*, LNCS, **2090**, pp. 278–314. Springer-Verlag.

[9] Harrison, P. G. (2003) Turning back time in Markovian process algebra. *Theoretical Computer Science*, **290**, 1947–1986.

[10] Hermanns, H., Kwiatkowska, M., Norman, G., Parker, D., and Siegle, M. (2003) On the use of MTBDDs for performability analysis and verification of stochastic systems. *Journal of Logic and Algebraic Programming*, **56**, 23–67.

[11] Kuntz, M., Siegle, M., and Werner, E. (2004) Symbolic performance and dependability evaluation with the tool caspa. *EPEW'04, Proceedings of European Performance Evaluation Workshop (FORTE satellite workshop)*, October, Lecture Notes in Computer Science, **3236**, pp. 293–307. Springer.

[12] Hillston, J. and Kloul, L. (2001) An efficient Kronecker representation for PEPA models. *Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, September, Lecture Notes in Computer Science, **2165**, pp. 120–135. Springer.

[13] Thomas, N. and Zhao, Y. (2011) Mean value analysis for a class of PEPA models. *The Computer Journal*, **54**, 643–652.

[14] Hillston, J. (2005) Fluid flow approximation of PEPA models. *QEST'05, Proceedings of the 2nd International Conference on Quantitative Evaluation of Systems*, Torino, September, pp. 33–42. IEEE Computer Society Press.

[15] Sumpter, D. J. T. (2000) From Bee to Society: An Agent-based Investigation of Honey Bee Colonies. PhD thesis Department of Mathematics, University of Manchester Institute of Science and Technology.

[16] Sumpter, D. J. T. and Broomhead, D. S. (2001) Relating individual behaviour to population dynamics. *Proceedings of the Royal Society: Series B*, **268**, 925–932.

[17] Benaïm, M. and Le Boudec, J.-Y. (2008) A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, **65**, 823–838.

[18] Massoulié, L. and Vojnovic, M. (2008) Coupon replication systems. *IEEE/ACM Transactions on Networking*, **16**, 603–616.

[19] Bakhshi, R., Cloth, L., Fokkink, W., and Haverkort, B. (2009) Mean-field analysis for the evaluation of gossip protocols. *QEST'09, Proceedings of the 5th IEEE Conference on the Quantitative Evaluation of Systems*, September, pp. 247–256. IEEE Computer Society.

[20] Ganesh, A., Lilienthal, S., Manjunath, D., Proutiere, A., and Simatos, F. (2010) Load balancing via random local search in closed and open systems. *ACM SIGMETRICS Performance Evaluation Review*, **38**, 287.

[21] Shakkottai, S. and Johari, R. (2010) Demand-aware content distribution on the Internet. *IEEE/ACM Transactions on Networking*, **18**, 476–489.

[22] Bakhshi, R., Cloth, L., Fokkink, W., and Haverkort, B. R. (2011) Mean-field framework for performance evaluation of push-pull gossip protocols. *Performance Evaluation*, **68**, 157–179.

[23] Bortolussi, L. and Policriti, A. (2007) Stochastic concurrent constraint programming and differential equations. *QAPL'07, 5th Workshop on Quantitative Aspects of Programming Languages*, September, Electronic Notes in Theoretical Computer Science, **190**, pp. 27–42. Elsevier.

[24] Cardelli, L. (2008) On process rate semantics. *Theoretical Computer Science*, **391**, 190–215.

[25] Hayden, R. and Bradley, J. T. (2010) A fluid analysis framework for a Markovian process algebra. *Theoretical Computer Science*, **411**, 2260–2297.

[26] Tribastone, M., Gilmore, S. T., and Hillston, J. (2010) Scalable differential analysis of process algebra models. *IEEE Transactions on Software Engineering*, **99**.

[27] McCaig, C., Norman, R., and Shankland, C. (2011) From individuals to populations: A mean field semantics for process algebra. *Theoretical Computer Science*, **412**, 1557–1580.

[28] Bortolussi, L. (2008) On the approximation of stochastic Concurrent Constraint Programming by master equation. *QAPL'08, Sixth Workshop on Quantitative Aspects of Programming Languages*, December, Electronic Notes in Theoretical Computer Science, **220**, pp. 163–180. Elsevier.

[29] Clark, A., Duguid, A., Gilmore, S. T., and Tribastone, M. (2008) Partial evaluation of PEPA models for fluid-flow analysis. In Thomas, N. and Juiz, C. (eds.), *Proceedings of the 5th European Performance Engineering Workshop on Computer Performance Engineering (EPEW)*, August, Lecture Notes in Computer Science, **5261**, pp. 2–16. Springer Berlin Heidelberg.

[30] Hayden, R., Stefanek, A., and Bradley, J. T. (2011) Fluid computation of passage time distributions in large Markov models. *Theoretical Computer Science*, **10.1016/j.tcs.2011.07.017**. (In Press).

[31] Stefanek, A., Hayden, R., and Bradley, J. T. (2011) Capturing the energy–performance trade-off in virtualised computing models. Technical report. Dept. of Computing, Imperial College London. `http://pubs.doc.ic.ac.uk/virtualised-energy/`. (Under review).

[32] Hayden, R. A. and Bradley, J. T. (2010) Evaluating fluid semantics for passive stochastic process algebra cooperation. *Performance Evaluation*, **67**, 260–284.

[33] Ciardo, G., Marie, R. A., Sericola, B., and Trivedi, K. S. (1990) Performability analysis using semi-Markov reward processes. *IEEE Transactions on Computers*, **39**, 1251–1264.

[34] Telek, M. and Rácz, S. (1999) Numerical analysis of large Markovian reward models. *Performance Evaluation*, **36–37**, 95–114.

[35] Stefanek, A., Hayden, R., and Bradley, J. T. (2011) Fluid analysis of energy consumption using rewards in massively parallel Markov models. *ICPE 2011, 2nd ACM/SPEC International Conference on Performance Engineering, March 14-16, 2011, Karlsruhe, Germany*, March, pp. 121–131. ACM.

[36] Chen, W., Huang, D., Kulkarni, A. A., Unnikrishnan, J., Zhu, Q., Mehta, P., Meyn, S., and Wierman, A. (2009) Approximate dynamic programming using fluid and diffusion approximations with applications to power management. *CDC 2009, Proceedings of the 48th IEEE Conference on Decision and Control*, December, pp. 3575–3580. IEEE.

[37] Stefanek, A., Hayden, R., and Bradley, J. T. (2011) Fluid computation of the performance-energy trade-off in large scale Markov models. *SIGMETRICS Performance Evaluation Review*, `http://pubs.doc.ic.ac.uk/fluid-performance-energy/`. (To appear).

[38] Bradley, J. T., Gilmore, S. T., and Hillston, J. (2008) Analysing distributed internet worm attacks using continuous state-space approximation of process algebra models. *Journal of Computer and System Sciences*, **74**, 1013–1032.

[39] Zhao, Y. and Thomas, N. (2010) Efficient solutions of a PEPA model of a key distribution centre. *Performance Evaluation*, **67**, 740–756.

[40] Clark, A., Gilmore, S., and Tribastone, M. (2009) Scalable analysis of scalable systems. *FASE'09, 12th International Conference on Fundamental Approaches to Software Engineering*, March, Lecture Notes in Computer Science, **5503**, pp. 1–17. Springer-Verlag.

[41] Massink, M., Latella, D., Bracciali, A., and Harrison, M. (2010) A scalable fluid flow process algebraic approach to emergency egress analysis. In Fiadeiro, J. L., Gnesi, S., and Maggiolo-Schettini, A. (eds.), *8th IEEE International Conference on Software Engineering and Formal Methods*, November, pp. 169–180. IEEE Computer Society.