Fluid-flow solutions in PEPA to the state space explosion problem

Richard A. Hayden Jeremy T. Bradley*

Abstract

Achieving the appropriate performance requirements for computer–communication systems is as important as the correctness of the end-result. This is particularly difficult in the case of massively parallel computer systems such as the clusters of PCs behind the likes of Google and peer-to-peer filesharing networks such as Bittorrent. Measuring the performance of such systems using a mathematical model is invariably computationally intensive. Formal modelling techniques make possible the derivation of such performance measures but currently suffer from the *state-space explosion problem*, that is, models become intractably large even for systems of apparently modest complexity. This work develops a novel class of techniques aimed at addressing this problem by approximating a representation of massive state spaces as more computationally-tractable real variables ('fluid-flow analysis').

1 Short introduction to performance modelling

Accurate performance modelling at the system design stage has never been more important than in a technological age dominated by large and complex computer and communication networks. Measurements such as request throughput or server utilisation can be used, for example, to predict the location of bottlenecks in the passage of requests through a computer network, and suggest steps to improve the situation.

A very useful mathematical tool for modelling many classes of systems is the *continuous-time Markov chain* (CTMC). CTMCs model the behaviour of a system by describing the set of possible states a system may be in and how the system moves between states over time. Models can be formalised directly into CTMCs, however there are many advantages to modelling a system using a higher-level abstraction, such as a *stochastic process algebra* (SPA) (e.g. PEPA [1], MTIPP [2], SPADES [3] and EMPA [4]), *stochastic Petri net* (SPN) [5, 6, 7] or *queueing network* [8, 9, 10, 11, 12]. Commonly, the model may be 'solved' through the computation of the steady-state analysis of an underlying CTMC[1] ('Markovian' formalisms). This project focuses on the well-known and very

*Dept. of Computing, Imperial College London, UK Email: {rah03,jb}@doc.ic.ac.uk

[1] In general, this requires the diagonalisation of a matrix with dimension equal to the number of states in the CTMC.

$$
\begin{aligned}
Processor_0 &\stackrel{def}{=} (task_1,\, r_1).Processor_1 \\
Processor_1 &\stackrel{def}{=} (task_2,\, q).Processor_0 \\
Resource_0 &\stackrel{def}{=} (task_1,\, r_2).Resource_1 \\
Resource_1 &\stackrel{def}{=} (reset,\, s).Resource_0 \\
Sys &\stackrel{def}{=} Processor_0 \underset{\{task_1\}}{\bowtie} Resource_0
\end{aligned}
$$

Fig. 1. Simple PEPA model of a processor and a resource

popular SPA, PEPA (*Performance Evaluation Process Algebra*), however the contributions are readily extensible to other Markovian formalisms.

Despite the relative tractability of CTMCs, models of realistic complexity can easily result in underlying state spaces of computationally intractable size. This phenomenon is known as 'state-space explosion' and is the current bottle-neck in the field of performance analysis, limiting the size of models and thus the complexity of systems that can be efficiently analysed.

Naturally, the demand for more accurate and finer-grained models of larger systems increases constantly, so there has been much research aimed at suppressing this explosion in some sense [13, 14, 15]. However, none of these approaches have succeeded in general at delivering the exponential speedup that is required to suppress this problem; the situation clearly demands a novel direction.

In this paper, we provide an overview and discussion of the work carried out in [16] which undertook a detailed analysis of the fluid-flow approximation techniques available to PEPA and other Markovian SPAs.

2 Overview of fluid-flow approximation

As discussed, this project focuses on the stochastic process algebra, PEPA. For a detailed explanation of PEPA, see [16, 1]. Fig. 1 is a simple PEPA model of a processor–resource system, where a single processor is communicating with a single resource over a shared $task_1$ action. This cooperation is shown by the last line of the model containing the $\underset{\{task_1\}}{\bowtie}$ symbol.

Fig. 2 shows the natural interpretation of this model as a *labelled transition system*, presenting the possible states of the Sys model and the rates at which it moves between said states.[2]

An extended model is perhaps one of a number of processors competing in parallel for a number of resources. To this end, we might adapt the definition

[2] If the reader is familiar with continuous-time Markov chains, they may also interpret this labelled transition system naturally as a CTMC (termed the *underlying CTMC of the model*).

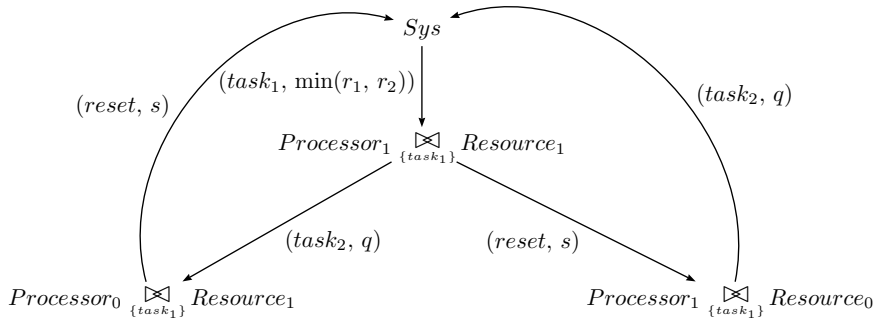

Fig. 2. Underlying CTMC of the PEPA model of Fig. 1

of the Sys model:

$$Sys \stackrel{def}{=} \underbrace{(Processor_0 \parallel \ldots \parallel Processor_0)}_{N_p} \underset{\{task_1\}}{\bowtie} \underbrace{(Resource_0 \parallel \ldots \parallel Resource_0)}_{N_r} \tag{1}$$

The $\parallel$ syntax is shorthand for $\underset{\emptyset}{\bowtie}$, i.e. pure concurrency with no synchronisation. This system consists of N_p $Processor_0$ components operating in parallel with N_r available $Resource_0$ components.

The simplicity of this model is unfortunately not reflected in the size of the underlying state space. Indeed, the number of states in the underlying CTMC is $2^{N_p+N_r}$ (i.e. grows exponentially with the size of the model) putting it very quickly out of the reach of traditional methods of analysis. The key to fluid-flow analysis is to stop tracking the state of every individual processor and resource, but instead, to keep track of *how many* are in each component state. This leads to a *natural aggregation* of the underlying state space and transition rates. For example, all of the following (N_p many) states:

$$\underbrace{(Processor_1 \parallel Processor_0 \parallel \ldots \parallel Processor_0)}_{N_p} \underset{\{task_1\}}{\bowtie} \underbrace{(Resource_0 \parallel \ldots \parallel Resource_0)}_{N_r}$$

$$\underbrace{(Processor_0 \parallel Processor_1 \parallel \ldots \parallel Processor_0)}_{N_p} \underset{\{task_1\}}{\bowtie} \underbrace{(Resource_0 \parallel \ldots \parallel Resource_0)}_{N_r}$$

$$\vdots$$

$$\underbrace{(Processor_0 \parallel Processor_0 \parallel \ldots \parallel Processor_1)}_{N_p} \underset{\{task_1\}}{\bowtie} \underbrace{(Resource_0 \parallel \ldots \parallel Resource_0)}_{N_r}$$

represent 'one $Processor_1$, N_p-1 $Processor_0$, zero $Resource_1$ and N_r $Resource_0$ components'. We may combine all these states into one aggregate state, e.g. $((N_p - 1, 1), (N_r, 0))$. Each position in this 4-tuple counts the number of the respective component in each state. Fig. 3 shows the complete aggregated state space of Eq. (1) with $N_p = N_r = 2$, exemplifying how the transition rates are aggregated. Fig. 4 shows the transitions into and out of an arbitrary central state in the aggregated state space of the general model of Eq. (1).

Note that this aggregation alone reduces the size of the underlying state space,

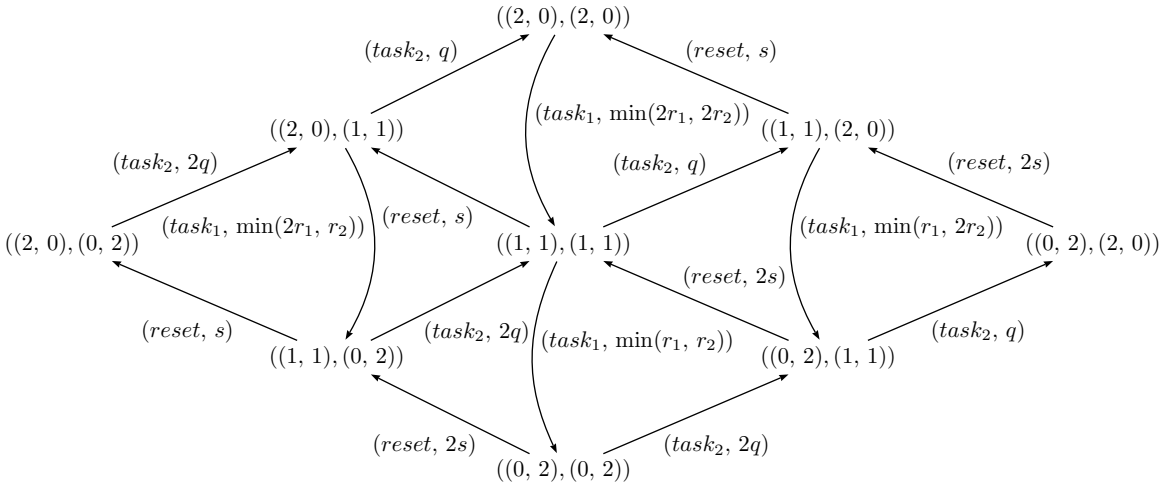

Fig. 3. Aggregated state space of the 2-processor/2-resource instance of (1)

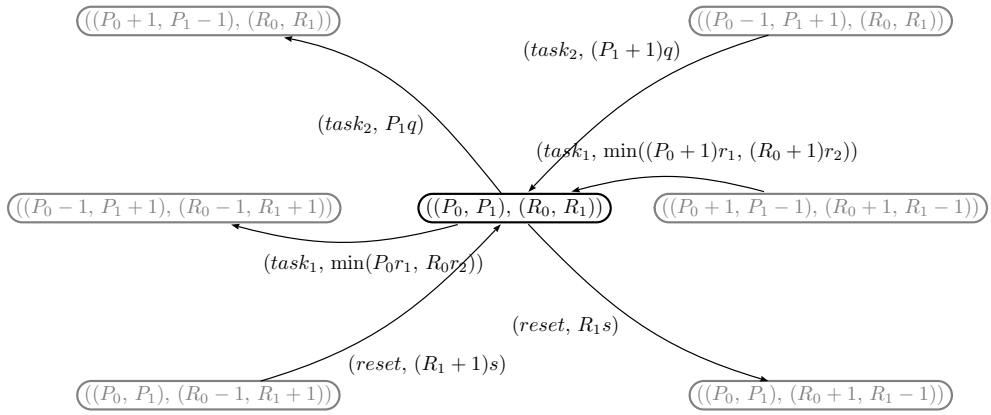

Fig. 4. A central state of the underlying aggregated CTMC of Eq. (1). The variable P_0 counts the number of $Processor_0$ components with P_1, R_0 and R_1 defined similarly.

but not to the extent that the state space explosion problem is eliminated[3].

The key point to note is that by aggregating in this manner, we have introduced integer counters into the model (counting the number of components in each possible state). It is these discrete values we desire to approximate as continuous variables. Such a fluid-flow approximation was first suggested by [17] (later extended by [18]), presenting a set of coupled *ordinary differential equations* (ODEs) naturally associated with a PEPA model whose solutions at time t were to be interpreted as continuous approximations to these discrete quantities. The argument of [18] is purely heuristic[4] and leads to the following system of coupled first-order ODEs:

$$\begin{aligned}
\frac{dP_0(t)}{dt} &= P_1(t)q - \min(P_0(t)r_1,\, R_0(t)r_2) \\
\frac{dP_1(t)}{dt} &= \min(P_0(t)r_1,\, R_0(t)r_2) - P_1(t)q \\
\frac{dR_0(t)}{dt} &= R_1(t)s - \min(P_0(t)r_1,\, R_0(t)r_2) \\
\frac{dR_1(t)}{dt} &= \min(P_0(t)r_1,\, R_0(t)r_2) - R_1(t)s
\end{aligned}$$

Elementary theory of first-order ODEs[5] allows us to deduce that they have a unique solution which is easily found by numerically integrating them. Fig. 5 compares the ODE solution with the traditional 'steady-state' solution of the underlying CTMC for initial conditions of 40 $Resource_0$ components and 100 $Processor_0$ components. We observe empirically that the results agree for the long-term expected state.

The model of Fig. 6 is an example of a system which is not as amenable to fluid approximation, because of its internal cooperation. Applying the same heuristic techniques yields the following sets of ODEs:

$$\begin{aligned}
\frac{dA_0(t)}{dt} &= \mu A_1(t) - \lambda N_A \mathbf{I}'(A_1(t)) \\
\frac{dA_1(t)}{dt} &= -\mu A_1(t) + \lambda N_A \mathbf{I}'(A_1(t))
\end{aligned}$$

where $\mathbf{I}'(.) := 1 - \mathbf{I}(.)$ is the complement of the indicator function:

$$\mathbf{I}(x) := \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$

Numerical integration of such ODEs is very questionable. The term involving the indicator function only has an effect if $A_1(t)$ ever reaches zero. Indeed, if one is to integrate these ODEs numerically, this does not actually ever happen, $A_1(t)$ merely tends towards zero (see Fig. 7). Compare this with the expectations obtained through stochastic simulation (Fig. 8).

[3] Indeed, it basically means the size of the state space grows exponentially with the number of states each individual processor and resource can be in, as opposed to the number of processors and resources in the parallel cooperation.

[4] It involves considering the change in the number of components in a particular state over a small time period δt, for example for P_0, they construct $P_0(t+\delta t) - P_0(t) = P_1(t)q\delta t - \min(P_0(t)r_1,\, R_0(t)r_2)\delta t$ and take the limit as $\delta t \to 0$.

[5] Specifically, the Picard-Lindelöf theorem.

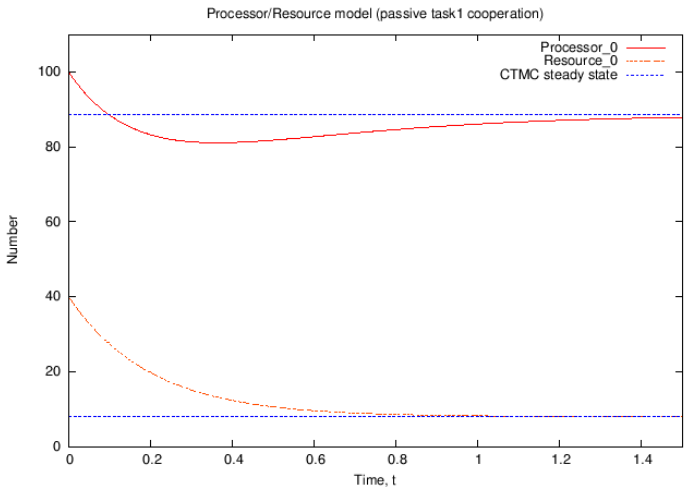

Fig. 5. Comparison of ODE solutions with steady state solutions of the underlying CTMC for simple processor/resource model.

$$
\begin{aligned}
A_0 &\stackrel{def}{=} (a,\, \lambda).A_1 \\
A_1 &\stackrel{def}{=} (b,\, \mu).A_0 \\
Sys &\stackrel{def}{=} \underbrace{\left(A_0 \underset{\{a\}}{\bowtie} A_0 \underset{\{a\}}{\bowtie} \ldots \underset{\{a\}}{\bowtie} A_0\right)}_{N_A}
\end{aligned}
$$

Fig. 6. Simple PEPA model with internal cooperation

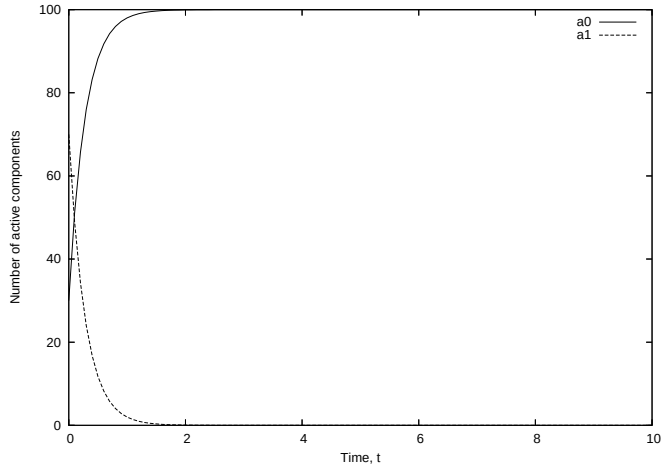

Fig. 7. ODE solution of the PEPA model of Fig. 6

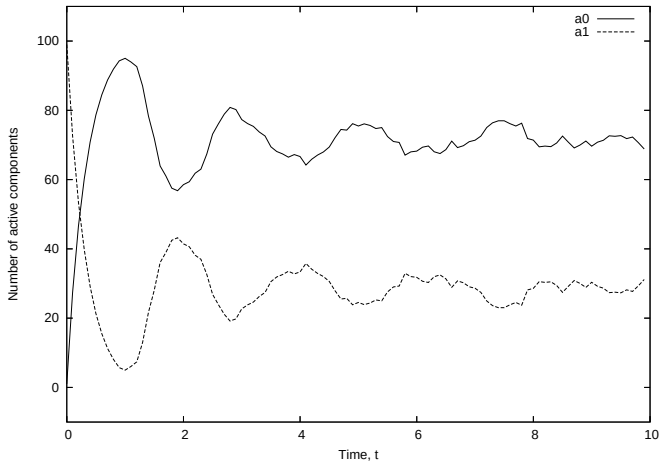

Fig. 8. Stochastic simulation of the PEPA model of Fig. 6

The results clearly do not agree. Indeed, the work of [18] is only justified by a heuristic argument and although empirically promising in many cases, is clearly invalid in many others.

3 Further development of fluid-flow analysis

3.1 ODE formalisation

The first contribution of this work was to provide a formal mathematical framework for deriving such systems of ODEs from a restricted class of PEPA models, allowing us to establish the limitations of this ODE technique and to motivate the development of more powerful methods in the latter part of the work. To do this, we associate with each 'transition class' a *transition rate function*, giving the rate of that particular transition. This is best seen by example: the transition rate functions for the model of Eq. (1) whose central state is shown in Fig. 4 are:

$$\begin{aligned} f_{-1,1,-1,1}(x_1, x_2, x_3, x_4) &= \min(x_1 r_1, x_3 r_2) \\ f_{1,-1,0,0}(x_1, x_2, x_3, x_4) &= x_2 q \\ f_{0,0,1,-1}(x_1, x_2, x_3, x_4) &= x_4 s \end{aligned} \tag{2}$$

The subscripts on the functions define the 'transition class', i.e. the amount by which each quantity changes due to such a transition.

The key is to define the *joint moment generating function* for the component state counting variables, e.g. for the model of Eq. (1), we define:

$$M(\theta_1, \theta_2, \theta_3, \theta_4, t) := \mathbf{E}[e^{P_0(t)\theta_1 + P_1(t)\theta_2 + R_0(t)\theta_3 + R_1(t)\theta_4}]$$

We then proceed to derive the following result formally from the underlying

CTMC:

$$\frac{\partial M}{\partial t} = \sum_{i_1,\ldots,i_N \in \mathbb{Z}} \left(e^{\theta_1 i_1 + \ldots + \theta_N i_N} - 1\right) f_{i_1,\ldots,i_N}\left(\frac{\partial}{\partial \theta_1}, \ldots, \frac{\partial}{\partial \theta_N}\right) M(\theta_1, \ldots, \theta_N, t) \tag{3}$$

However, this of course only makes sense where the transition rate functions are polynomials. This rules out transition rate functions involving the $\min(\cdot,\cdot)$ function (e.g. the model of (1)) or indicator functions (e.g. the model of Fig. 6). This would suggest that this justification only holds for models not involving the $\underset{A}{\bowtie}$ combinator where $A \neq \emptyset$.

In a similar fashion, we are also able to deal with passive cooperation (a popular PEPA synchronisation type) using the generating function technique and this too agrees with the heuristic presentation of [18]. Indeed, we prove for a restricted class of PEPA models (approximately those involving no internal cooperation and only passive external cooperation) that the ODEs of [18] are correct when interpreted in terms of the stochastic expected values of the state variables [16, Sections 4.2.2, 4.2.3]. We were also able to generalise this to derive similar systems of ODEs for higher-order moments such as the variance (not attempted even heuristically by [18]) for the restricted class of models [16, Section 4.3]. This is very important because the expectation can be misleading without also having knowledge of the variance.

Having introduced this formal framework, we were able to identify the key limitations of such an ODE-based fluid-flow analysis [16, Section 4.4]:

1. It is only formally-justified for a restricted class of PEPA models — there is no guarantee of correctness for more general models (even if results seem to be empirically correct) and no (even empirical) extension to higher-order moments for models outside of the restricted class.

2. It is not even empirically valid for 'discrete' models (e.g. those involving internal cooperation).

It was clear more powerful mathematical techniques were required to develop further the applicability of fluid-flow analysis to performance modelling.

3.2 Fluid-flow analysis with SDEs

A stochastic differential equation (SDE) is a differential equation in which one or more of the terms is a stochastic process, thus resulting in a solution which is itself a stochastic process. Commonly, SDEs are used in situations where it is desired to model the effect of the perturbation of a system of ODEs by random noise. Indeed, re-introducing stochastic non-determinism into the ODEs detailed in the previous section would seem like a potentially fruitful avenue to develop the fluid-flow analysis techniques further.

Brownian motion is the fundamental continuous-time stochastic process in terms of which most SDEs are formulated. It is closely related to the normal distribution and can be used to develop *functional central limit theorems* for stochastic

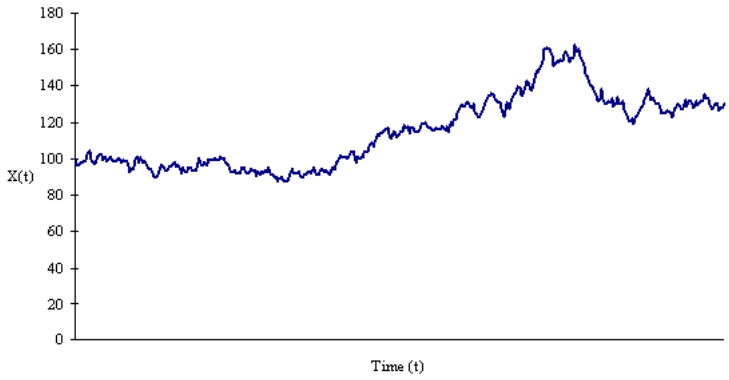

Fig. 9. A trace of a Brownian motion started at $X(0) = 100$.

processes in the same way normal random variables can be used to approximate other random variables (central limit theorems). However, it shares some difficult mathematical properties,[6] which demand a new calculus (the Itô calculus). Fig. 9 shows a trace of a Brownian motion.

The aim of our work here was to develop a functional central limit theorem for PEPA models using Brownian motions to approximate certain stochastic processes that fully determine the PEPA model in question. Indeed, the stochastic processes we identified were the processes counting the number of each class of transitions.[7] We termed these *transition counting processes* [16, Section 5.1]. Recall the transition rate functions Eq. (2) for the model of Eq. (1). In the order of enumeration of Eq. (2), let the associated transition counting processes be $N_1(t)$, $N_2(t)$ and $N_3(t)$. These clearly fully determine the values of $P_0(t)$, $P_1(t)$, $R_0(t)$ and $R_1(t)$ as:

$$\begin{aligned} P_0(t) &= P_0(0) - N_1(t) + N_2(t) \\ P_1(t) &= P_1(0) + N_1(t) - N_2(t) \\ R_0(t) &= R_0(0) - N_1(t) + N_3(t) \\ R_1(t) &= R_1(0) + N_1(t) - N_3(t) \end{aligned}$$

For high transition rates, we were then able to show that these discrete processes converge in probability distribution to an integral with respect to a Brownian motion [16, Section 5.2].[8] This holds even for transition rate functions involving $\min(\cdot, \cdot)$ functions and since it is an approximation to the stochastic trace of the actual system easily facilitates calculation of higher-order moments too.

[6] For example, Brownian motion is almost always nowhere differentiable and not monotone on almost every interval no matter how small it is.

[7] *Class* in the sense of the previous section – that is, these stochastic processes are in one-to-one correspondence with transition rate functions.

[8] Explicitly, we use the martingale convergence techniques of [19].

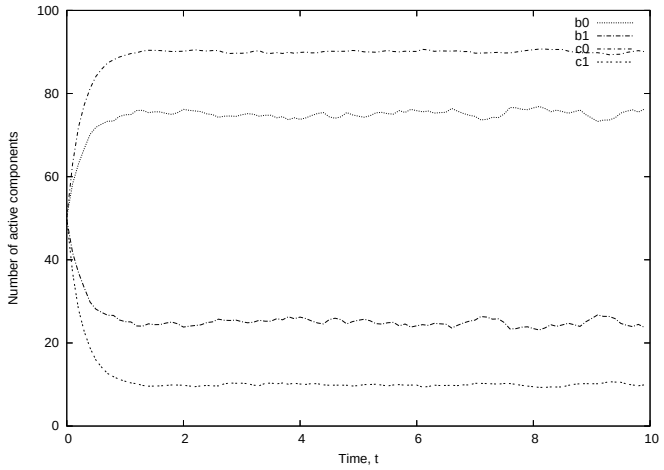

Fig. 10. Expectations via stochastic simulation of the PEPA model a hybrid discrete–continuous system.

However, we are still unable to deal with systems involving pathologically slow rates (e.g. those involving indicator functions such as that of Fig. 6) since this approximation is only valid in the limit of high transition rates. Indeed the model of Fig. 6 is never going to be amenable to a fluid-flow approximation because it involves only 'discretely-behaving' components. However some models may involve both components which behave in this discrete manner and components behaving in a more fluid-like manner.

A hybrid scheme where we may choose to apply a functional central limit theorem only to a a subset of the transition counting processes and leave the rest to be treated as discrete processes is clearly desired. However, such models involving 'jump processes' with continuously-varying state-dependent rates are known to be generally intractable. That said, similar models arise in the analysis of LIBOR[9] financial markets and the author was able to adapt work in this area [20, 21] to the context of Markovian modelling formalisms [16, Chapter 6], using a technique known as *thinning a Poisson random measure*.

Using this Poisson thinning technique, we have been able to successfully recreate the time evolution of hybrid discrete–continuous models. To briefly demonstrate this: Fig. 10 shows a set of the actual traces obtained by stochastic simulation for a hybrid model; Fig. 11 shows the same traces when we try to calculate the same expectations using the original approximations of [18]; finally the hybrid fluid-modelling scheme, using Poisson thinning, can be seen to produce more realistic traces in Fig. 12.

[9]London Interbank Offered Rate.

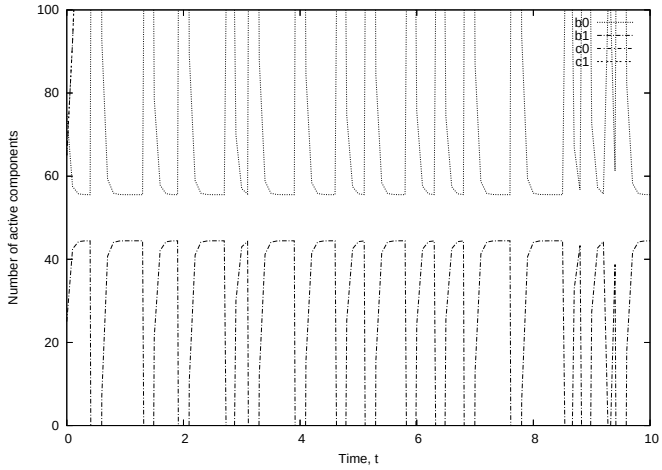

Fig. 11. Expectations via ODEs of the PEPA model of a hybrid discrete–continuous system.

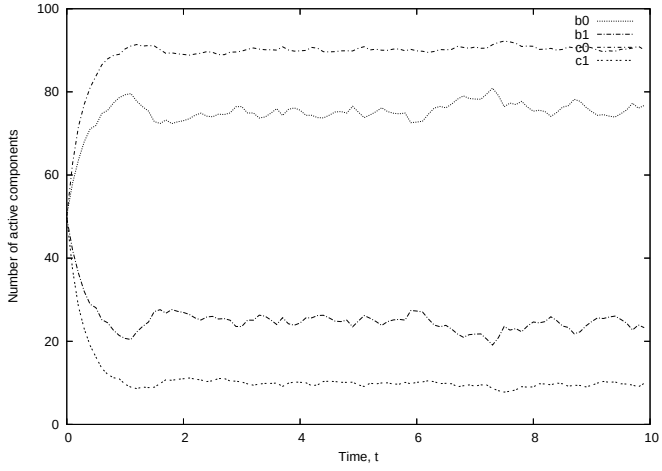

Fig. 12. Expectations via SDEs of the PEPA model of a hybrid discrete–continuous system.

4 Conclusions

This paper provides a synopsis for the broader project presented in [16] on the subject of fluid-flow analysis of PEPA models.

We provide a formal mathematical framework for generating systems of coupled first-order ODEs from certain Markovian modelling formalisms, substantially extending the initial work of [18]. In the explicit case of a subset of PEPA models, we provide an interpretation and a proof of correctness. We identify the mathematical reasons preventing more general PEPA models being analysed in this fashion, delivering substantial insight into the amenability of PEPA itself to fluid-flow analysis. We also identify the key limitations of the analysis. [16, Sections 4.2.1, 4.2.2, 4.2.3, 4.4]

We extend the work of [18] to the generation of similar systems of coupled ODEs for higher-order moments such as the variance, for which no method was previously available. [16, Section 4.3]

We show formally how PEPA models can be recast as a system of coupled stochastic differential equations (SDEs), by using a novel decomposition of the PEPA model into *transition counting processes*. This allows us to perform fluid-flow analysis on a larger class of PEPA models. [16, Sections 5.1, 5.2, 5.3]

By adapting the work of [20, 21] on certain classes of financial models, we are able to present a hybrid scheme consisting of systems of SDEs driven by both Brownian motions and thinned Poisson random measures. Using this approach, components amenable to continuous state space approximation may be analysed while maintaining discretely-behaving components in non-fluid form. Such an approach is entirely novel — up until now, a continuous state space approximation could either be applied to the whole model (leading to poor results should it include some discretely-behaving components) or the whole model had to be dealt with discretely (and thus potentially incurring the state space explosion problem). [16, Chapter 6]

References

[1] J. Hillston, *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[2] U. H. N. Gotz and M.Rettelbach, "TIPP - a language for timed processes and performance evaluation," in *1st Workshop on Process Algebra and Performance Modelling*, University of Edinburgh, May 1993.

[3] P. Harrison and B. Strulo, "SPADES - a process algebra for discrete event simulation," *Journal of Logic and Computation*, vol. 10, pp. 3–42, January 2000.

[4] M. Bernardo and R. Gorrieri, "A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time," *Theor. Comput. Sci.*, vol. 202, no. 1-2, pp. 1–54, 1998.

[5] M. K. Molloy, "Performance analysis using stochastic Petri nets," pp. 268–272, 1990.

[6] S. Natkin, *S. Reseaux de Petri Stochastiques*. PhD thesis, CNAM-PARIS, 1980.

[7] S. Donatelli, "Superposed generalized stochastic Petri nets: Definition and efficient solution," in *Proceedings of the 15th International Conference on Application and Theory of Petri Nets*, (London, UK), pp. 258–277, Springer-Verlag, 1994.

[8] J. W. Cohen, *The Single Server Queue*, vol. 8 of *Applied Mathematics and Mechanics*. North-Holland, 1969.

[9] D. G. Kendall, "Some problems in the theory of queues," *Journal of the Royal Statistical Society*, vol. B–13, no. 2, pp. 151–185, 1951.

[10] J. Keilson, "The ergodic queue length distribution for queueing systems with finite capacity," *Journal of the Royal Statistical Society*, vol. B, no. 28, pp. 190–201, 1966.

[11] L. Kleinrock, *Queueing Systems, Volume I: Theory*. John Wiley & Sons, 1975.

[12] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications*. John Wiley & Sons, 1976.

[13] D. D. Deavours and W. H. Sanders, "An efficient disk-based tool for solving very large Markov models," in *Computer Performance Evaluation*, pp. 58–71, 1997.

[14] W. Knottenbelt, *Parallel Performance Analysis of Large Markov Models*. PhD thesis, Department of Computing, Imperial College London, 1999.

[15] P. G. Harrison, "Turning back time in Markovian process algebra," *Theoretical computer science*, vol. 290, pp. 1947–1986, 2003.

[16] R. A. Hayden, "Addressing the state space explosion problem for PEPA models through fluid-flow approximation." Undergraduate dissertation. Dept. of Computing, Imperial College London, June 2007.

[17] J. Hillston, "Fluid flow approximation of PEPA models," in *Proceedings of the Second International Conference on the Quantitative Evaluation of Systems*, (Torino, Italy), pp. 33–43, IEEE Computer Society Press, Sept. 2005.

[18] J. T. Bradley and J. Hillston, "Using ODEs and continuous state-space approximation to analyse massively parallel process algebra models," in *Journal of Software and Systems*, 2007. (submitted).

[19] T. Kurtz and S. Ethier, *Markov Processes - Characterisation and Convergence*. Wiley, 1986.

[20] P. Glasserman and N. Merener, "Numerical solution of jump-diffusion LIBOR market models."

[21] P. Glasserman and N. Merener, "Convergence of a discretization scheme for jump-diffusion processes with state-dependent intensities."