Fluid Analysis of Energy Consumption using Rewards in Massively Parallel Markov Models

Anton Stefanek Richard A. Hayden Jeremy T. Bradley

Department of Computing
Imperial College London
London SW7 2BZ
anton.stefanek@ic.ac.uk

ABSTRACT

Capturing energy consumption directly from a stochastic behavioural model is a computationally expensive process. Using a so-called fluid analysis technique we are able to access accumulated reward measures in much larger scale stochastic systems than has been previously possible. These accumulated rewards are ideal for deriving energy and power consumption from stochastic process models. In previous work, it has been shown how to derive a set of ordinary differential equations (ODEs) whose solutions approximate the moments of component counts in a continuous-time Markov chain (CTMC) described in a stochastic process algebra. In this paper, we show how to extend the method to provide rapid access to moments of *accumulated rewards* in CTMCs. In addition to measuring the amount of energy used by a system, we are also interested in the time taken to reach a particular level of energy consumption. In reward terms, this is a so-called completion time. In this paper, we are able to use higher moments of rewards to give us access to completion time distributions.

We demonstrate the technique on a model of energy consumption in a client–server system with server failure and hibernation. Moreover, we are able to use these new and rapid techniques to capture the trade-off between energy consumption and service level agreement (SLA) compliance. We use a standard optimisation approach to find the precise configuration of the system which minimises the energy consumption while satisfying an operational response-time quantile.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems—*Reliability, availability, and serviceability*; G.3 [**Probability and Statistics**]: Markov processes

General Terms

Performance

ICPE'11, March 14–16, 2011, Karlsruhe, Germany.

1. INTRODUCTION

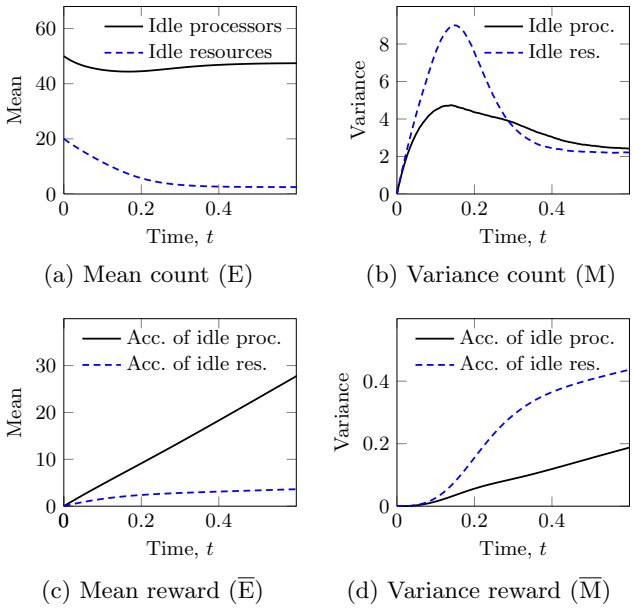
Energy consumption is a critical factor in the practical operation of massive computer systems on the internet today. Whether in wireless networks, virtualised services that run on Amazon's Elastic Compute Cloud or Google's search clusters, energy consumption, heat management and cost are required metrics. Capturing these features in a performance model has traditionally been achieved using Markov Reward Models (MRMs) where behavioural models have been augmented with random variables that accumulate a reward (such as energy or cost) for being in a particular state, or transitioning to a new state [1, 2, 3].

However, given the scale of a Google search cluster or a typical cloud environment, it would be impractical to perform traditional reward analysis on these massively parallel systems. A new technique is needed. Recently, so-called fluid analysis of performance models [4, 5] makes it possible to analyse systems which exhibit a high degree of parallelism. We also have the ability to generate higher moments in the course of such analysis and have a good understanding of when the first-order analysis produces a good transient approximation to the underlying Markov model [5, 6].

The contribution of this paper is to show, for the first time, how it is possible to derive reward measures from massive stochastic models using fluid analysis techniques. We show that many measures can be constructed precisely as functions of existing component counts. Reward measures can be represented using ordinary differential equations (ODEs) that augment the set that already exists to analyse the behavioural system directly. We show that higher moments of reward measures can similarly be generated. We demonstrate how reward passage times or so-called *completion times* can be constructed and solved in terms of rewards, giving for instance the passage time until the temperature in a server room reaches a certain level given a particular server load profile. Finally, we illustrate the technique on a model of energy consumption in a client–server system with server failure and hibernation and use the fast reward analysis to assess the trade-off between energy consumption and service level agreement (SLA) compliance.

1.1 Motivating example

We first look at a typical reward example and we list the main quantities of interest that the techniques can provide. Consider the ubiquitous situation of m identical processors running in parallel, each in need of one of n identical resources. Each processor has two possible states and loops between them repeatedly. In the first state the processor



(a) Mean count (E)

(b) Variance count (M)

(c) Mean reward ($\overline{\text{E}}$)

(d) Variance reward ($\overline{\text{M}}$)

Figure 1: Selected plots of important quantities: (a) Mean component count, (b) Variance of component count, (c) Mean accumulated reward and (d) Variance of accumulated reward

is acquiring a resource, in the second state it returns without synchronisation to the first state. Each resource similarly has two possible states that it loops between. In the first state, the resource awaits acquisition from a processor, in the second it resets without synchronisation to the first state. The actions the system takes (e.g. a processor acquiring a resource or a resource resetting) are stochastic in nature. This gives us a model with four component states, which is defined formally in Section 3.

Ideally, in order to fully understand the behaviour of the system over time, we would like to obtain the distribution of the individual component counts at each point of time as well as at the point when the system runs for sufficiently long for the time not to have any influence (the *steady state* of the system). However, this can be too costly and sometimes simpler measures are sufficient for the modelling purposes. For example, to get a basic understanding of how the system evolves, we can look at the means of the four individual component counts at each point of time or as the time goes to infinity. To get insight into the variability in the underlying stochastic model, we can also derive the variance and higher moments of these counts [5].

We will be interested in *rewards*, quantities which are accumulated over time in some way, such as energy consumption or the total cost of running the system. At each point in time, these will depend on the complete history of the individual component counts. As in the case of the above measures, the mean values, variances and higher moments of these accumulated counts can serve our modelling purposes. The accumulated rewards may grow indefinitely with time and to provide any information about their values in the steady state, the rewards have to be normalised in some way. It is sometimes interesting to look at the rate of increase in the reward in the steady state: that is, to look at the limit of the reward divided by t as t goes to infinity.

The following list summarises the quantities we are interested in of which (E)–(P) can already be calculated by means of differential equation approximations, whereas ODE approximations of ($\overline{\text{E}}$)–($\overline{\text{P}}$) are the subject of investigation in this paper:

- (E) the *mean* number of components in each of the four states at any time $t \geq 0$
- (M) variance and other higher and joint *moments* of the counts
- (S) the mean component counts and moments in the *steady state* of the system (as $t \to \infty$)
- (P) the passage-time distribution of individual components or groups of components reaching a particular state
- ($\overline{\text{E}}$) the mean of the accumulation of the component counts until each time $t \geq 0$
- ($\overline{\text{M}}$) variance and other higher and joint moments of the accumulated quantities
- ($\overline{\text{S}}$) normalised accumulated quantities (divided by t) and their moments in the steady state of the system (as $t \to \infty$)
- ($\overline{\text{P}}$) the passage-time or completion-time distribution for a system to reach a particular reward level

Figure 1 shows these quantities for processors and resources in their respective initial states. Note that the plots in (a) and (b) remain constant after a longer period of time. These constant values correspond to the quantities (S). Similarly the plot in (c) is linear after some time and the (constant) value of its gradient corresponds to the quantity ($\overline{\text{S}}$). Plot (d) displays the variance of the reward which will be useful for gauging the stochastic reward precision.

All of the quantities (E)–($\overline{\text{P}}$) can be obtained via stochastic simulation of the system. However, with increasing initial counts of processors and resources the traditional simulation techniques become expensive as they have to cope with frequent events of short duration. Moreover, with increasing order of the moments of interest, the number of simulation replications needed greatly increases. For the quantities (E)–(S), this problem is addressed in [5] (based on the earlier work in [4]), where a system of ODEs is derived that approximates the temporal evolution of the means and higher and joint moments of the individual counts. Numerically solving these ODEs is computationally less expensive than the simulations and can thus provide fast access to the moments with a reasonable degree of accuracy. The nature of this approximation revolves around so-called *switch points* and is further investigated in [6]. In [7] it is shown how to use the moments to derive approximations to distributions of the time it takes for individual components or groups of components to reach a particular set of states, the quantities (P).

The main contribution of this work is an extension of the method from [5] that gives ODEs for the quantities ($\overline{\text{E}}$)–($\overline{\text{S}}$). We give an exact form of these ODEs and show how to apply similar approximations to those from [5] to get a system that can be numerically solved alongside the ODEs for moments of component counts. We show how to use the moment approximations to derive approximations to the distributions of completion times, the quantities ($\overline{\text{P}}$).

2. RELATED WORK

Stochastic Reward Models (SRMs) comprise a stochastic process $\{Z(t) : t \geq 0\}$ and a reward measure, $B(t)$. As the stochastic process changes state, reward is accumulated in the measure $B(t)$. In deterministic reward models (as considered in this paper), reward is gathered in a particular state proportional to the time spent in that state. Additionally reward can also be gained instantaneously as the system changes state through so-called impulse rewards. Reward models that are applied to Markov models give so-called Markov Reward Models (MRMs) and there is much prior work that analyses the underlying reward process [1, 2, 3, 8].

2.1 Reward Strategies

Different strategies for reward accumulation are shown in Figure 2. In Figure 2(a), reward is gained in state i at rate r_i, and on moving to state k, this is added to by a further reward accumulated at rate r_k. On changing state, reward is gained in a continuous fashion. There is no loss of reward although there is no reason why a state should not accumulate a negative reward with associated negative rate. This is called a Preemptive Resume (PRS) strategy for reward accumulation.

In Figure 2(b), the reward is lost on changing state with reward only accumulated linearly by the current state that the system is in. This is a Preemptive Restart (PRT) strategy, sometimes called Preemptive Repeat.

Although not considered here, another important reward strategy is the impulse reward. In this case an instantaneous reward addition is made on changing state [3]. Strategies involving partial loss of rewards at state change instants are also possible [9] but are not considered further here. A good summary of reward strategies can be found in Horváth *et al.* [10].

Preemptive Restart is fairly straightforward to implement and can be done traditionally using a reward measure vector on a transient or steady-state vector for a system. Preemptive Resume requires an integration over the stochastic process to capture the accumulation over many states. In this paper, we consider the Preemptive Resume reward strategy for massive Markov models comprising many parallel components. We show that, in particular, the Preemptive Resume rewards and higher moments can be formulated using additional differential equations to capture the reward over and above the ODEs used to analyse the massively parallel system.

Since we are working with component counts rather than individual components, in our setting, rewards have to be functions of those component counts. This certainly covers the traditional linear accumulation scenario, but also allows for the possibility of more complex rewards, such as ones involving a nonlinear combination of the component counts.

2.2 Completion times

One of the most useful aspects of rewards is the ability to express *completion times* in terms of the accumulated reward (see for instance [11]). For instance, we might wish to calculate the time to accumulate a reward of 20 MWh of energy in a power station model. In stochastic reward models, this is achieved in terms of a completion time distribution, $C(x)$:

$$C(x) = \inf\{t \geq 0 : B(t) > x\} \tag{2.1}$$

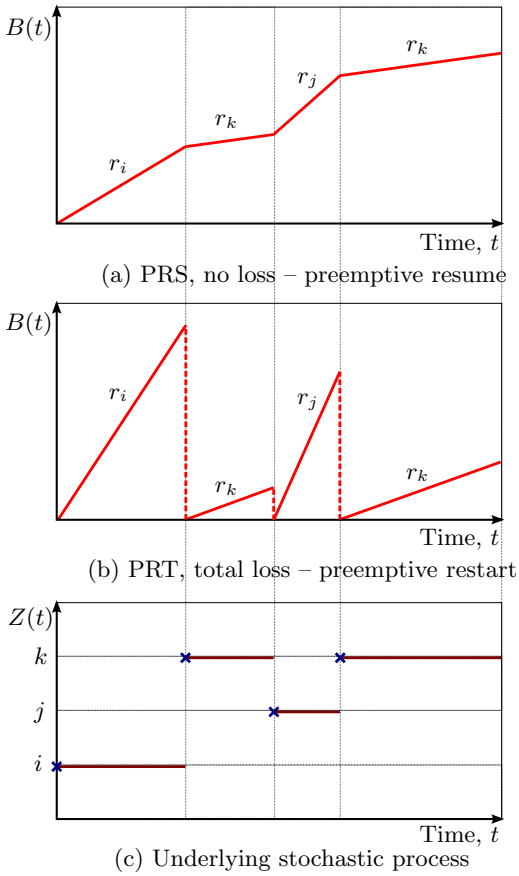


Figure 2: Change of accumulated reward, $B(t)$, for distinct reward strategies in (a) and (b) as the underlying process, $Z(t)$, evolves in (c)

being the first time that the system reward exceeds the value x. It is possible to derive bounds on this distribution using moments of the appropriate reward measures [10].

We will show how, through the use of reward moments derived from fluid analysis, we can also bound the distribution of the reward completion time for very large models.

2.3 Solution techniques

There has been a significant amount of research into efficient solution techniques for the distributions or moments of rewards and the completion times. Most of the techniques, including the approaches from [2, 3, 8, 11, 12, 13, 14], are based on numerical techniques which require explicit consideration of the entire state space of the associated discrete-state Markov model. A detailed overview can be found in [10].

To our best knowledge, all of the existing methods have complexity at least linearly dependent on the number of states. This suffers from the state space explosion problem, which is especially prominent in the massively parallel models we are interested in. In this paper we present a technique with complexity not dependent on the counts of the individual replicated components. A similar technique has been mentioned in the context of physical chemistry in [15]. However, it is limited only to components of a single type and therefore not applicable to our models of interest where components can be in multiple states.

3. GROUPED PEPA

Grouped PEPA or GPEPA [5] is a simple syntactic extension of the stochastic process algebra PEPA defined to provide a more elegant treatment of the ODE moment approximation. Formally, the extension introduces a further level in the syntax of PEPA, the *Grouped PEPA models*, defined as:

$$G ::= G \underset{L}{\bowtie} G \mid \mathbf{Y}\{P \parallel \cdots \parallel P\}$$

This defines a GPEPA model to be either a PEPA cooperation between two GPEPA models (over the set of actions L) or alternatively a labelled grouping of PEPA components, P, in parallel with each other. $\mathbf{Y}$ is the *group label*. The Grouped PEPA model is nothing more than a standard PEPA model with, additionally, a label to define the components involved in parallel grouping. These labels are used to define the level at which the ODE approximation to the system is made.

The example from Section 1.1 can be represented by the following GPEPA definition:

$$P_0 \stackrel{def}{=} (acquire,\, r_1).P_1 \qquad R_0 \stackrel{def}{=} (acquire,\, r_2).R_1$$
$$P_1 \stackrel{def}{=} (task,\, q).P_0 \qquad R_1 \stackrel{def}{=} (reset,\, s).R_0$$

$$System \stackrel{def}{=} \mathbf{Ps}\{P_0[m]\} \underset{\{acquire\}}{\bowtie} \mathbf{Rs}\{R_0[n]\}$$

3.1 Moment approximation via ODEs

Traditionally, the system states in PEPA models keep track of the state of each individual sequential component. This can lead to state space explosion, which makes the model not amenable to the standard analysis methods other than the computationally expensive stochastic simulation. The state space explosion is especially severe (with respect to the syntactical size of the model) in the case of models with groups consisting of many components acting in parallel. An established way to tackle this in the case of groups consisting of many identical components is by aggregating the state space by keeping track of *counts* of the individual components [4]. In the context of Grouped PEPA models, it is sufficient to represent each state of the underlying CTMC by a numerical vector $\mathbf{N}(t)$ consisting of counts $N_{G,P}(t)$ for each possible pair of group label G and component P (as in [5]). It has been shown in [5] how to derive approximations to the differential equations governing the expectations of these counts $\mathbb{E}[N_{G,P}(t)]$. For example, if we let $P_i(t)$ to stand for $N_{\mathbf{P},P_i}(t)$ and $R_i(t)$ for $N_{\mathbf{R},R_i}(t)$ for $i = 0, 1$, the method from [5] gives the approximations $\widetilde{\mathbb{E}}[\cdot]$ to the exact means $\mathbb{E}[\cdot]$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}[P_0(t)] = q\widetilde{\mathbb{E}}[P_1(t)] - \min(r_1\widetilde{\mathbb{E}}[P_0(t)], r_2\widetilde{\mathbb{E}}[R_0(t)])$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}[P_1(t)] = -q\widetilde{\mathbb{E}}[P_1(t)] + \min(r_1\widetilde{\mathbb{E}}[P_0(t)], r_2\widetilde{\mathbb{E}}[R_0(t)])$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}[R_0(t)] = s\widetilde{\mathbb{E}}[R_1(t)] - \min(r_1\widetilde{\mathbb{E}}[P_0(t)], r_2\widetilde{\mathbb{E}}[R_0(t)])$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}[R_1(t)] = -s\widetilde{\mathbb{E}}[R_1(t)] + \min(r_1\widetilde{\mathbb{E}}[P_0(t)], r_2\widetilde{\mathbb{E}}[R_0(t)])$$

The authors of [5] further extended this method to derive ODE approximations to higher and joint moments of the counts, such as variances, covariances and others. For example, we would have the approximation

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}&\widetilde{\mathbb{E}}[P_0(t)P_1(t)] \\ &= -\min(r_1\widetilde{\mathbb{E}}[P_0(t)], r_2\widetilde{\mathbb{E}}[R_0(t)]) \\ &\quad -\min(r_1\widetilde{\mathbb{E}}[P_0(t)P_1(t)], r_2\widetilde{\mathbb{E}}[R_0(t)P_1(t)]) \\ &\quad +\min(r_1\widetilde{\mathbb{E}}[P_0(t)^2], r_2\widetilde{\mathbb{E}}[R_0(t)P_0(t)]) \\ &\quad +q(\widetilde{\mathbb{E}}[P_1(t)^2] - \widetilde{\mathbb{E}}[P_1(t)] - \widetilde{\mathbb{E}}[P_0(t)P_1(t)]) \qquad (3.1)\end{aligned}$$

All of the above differential equations are approximations since the following approximation had to be applied to tackle the case of an expectation of a min expression:

$$\mathbb{E}[\min(X, Y)] \approx \min(\mathbb{E}[X], \mathbb{E}[Y]) \qquad (3.2)$$

In general, GPEPA models cause the right hand sides of the ODEs to additionally contain rational functions of the expectations. If they do, the model is called *splitting*, otherwise *split-free*. Further approximation is then caused by a sequence of nested applications of the above and of the approximation

$$\mathbb{E}[f(X_1, \ldots, X_k)] \approx f(\mathbb{E}[X_1], \ldots, \mathbb{E}[X_k]) \qquad (3.3)$$

for a rational function f.

The effects of the approximation Equation (3.2) on the error of the numerical solution to the system of ODEs were investigated in [6] with the help of a tool *Grouped PEPA Analyser* (GPA). In the following section we show how to derive ODEs for the quantities involving accumulated rewards. We also extend the GPA tool accordingly to apply the new technique to a larger model in Section 5.

4. APPROXIMATING MOMENTS OF REWARDS VIA ODES

In this section, we extend the above ODE approximation of GPEPA moments to include ODEs for the moments of the quantities $(\overline{\mathrm{E}})$–$(\overline{\mathrm{S}})$ from Section 1.1. We begin with the simplest case of the moments of accumulated rewards in the steady state and continue with the transient case.

We will calculate the quantities $(\overline{\mathrm{E}})$–$(\overline{\mathrm{S}})$ for the two following simple rewards. For the processors/resources model, we could be interested in the energy consumption of the processors, where say the state P_i consumes c_i units at each time instant, for $i = 0, 1$ respectively. As there are $P_i(t)$ components of P_i at each time t, for the total energy consumption we need to evaluate (means, moments, etc., of) the accumulated quantity

$$A_P(t) = \int_0^t c_0 P_0(u) + c_1 P_1(u)\mathrm{d}u \qquad (4.1)$$

This is the same kind of reward that would be expressible if we model each processor individually; in that case, we would evaluate a sum of m integrals with the corresponding indicator functions for each processor.

To illustrate a reward that could not be expressed in such a concise way, we could consider a quantity that uses a nonlinear function of the component counts:

$$A_{PR}(t) = \int_0^t c P_0(u) R_0(u)\mathrm{d}u \qquad (4.2)$$

This could represent a measure of total income if each processor costs c units for being able to use an available resource in each instant of time.

4.1 Steady state normalised rewards

It turns out that to access moments of accumulated rewards in the steady state of the system, the quantities ($\overline{\mathrm{S}}$), we can use the ODE moment method without any extensions. In order for the quantities to be finite in the steady state, we divide the reward by time, i.e. look at the rate of increase:

$$\frac{1}{t}\int_0^t X(u)\mathrm{d}u$$

as t goes to infinity. For the illustrative rewards A_P and A_{PR} we therefore need to evaluate the values of

$$A_P(t)/t \text{ and } A_{PR}(t)/t \text{ as } t \to \infty$$

We can express expectations of such quantities in general just by using the steady state limits of the means of the individual component counts $\mathbb{E}[P_i]$ and the moment $\mathbb{E}[P_0 R_0]$. This is a corollary, previously mentioned in [16], of a standard property of CTMCs:

THEOREM 4.1 ([17], THEOREM 3.8.1). *Let $\{\mathbf{X}(t) \in \mathbb{N}^k\}_{t\geq 0}$ be an irreducible, positive recurrent Markov process and $\mathbf{f}\colon \mathbb{N}^k \to \mathbb{R}$ a bounded function. Then:*

$$\mathbb{P}\left(\frac{1}{t}\int_0^t \mathbf{f}(\mathbf{X}(s))\mathrm{d}s \to \bar{\mathbf{f}}\right) = 1 \qquad \text{as } t \to \infty \tag{4.3}$$

where $\bar{\mathbf{f}} = \sum_{\mathbf{n}\in\mathbb{N}^k} \lambda_n \mathbf{f}(\mathbf{n})$ and $\lambda_{\mathbf{n}}$ is the unique invariant distribution.

Using this, we can directly get

$$\begin{aligned}\mathbb{E}\left[A_P(t)/t\right] &= c_0\mathbb{E}[P_0(t)] + c_1\mathbb{E}[P_1(t)]\\ \mathbb{E}\left[A_{PR}(t)/t\right] &= c\mathbb{E}[P_0(t)R_0(t)] \qquad \text{as } t\to\infty\end{aligned}$$

The method from [5] provides ODEs with solutions $\widetilde{\mathbb{E}}[\cdot]$ approximating the expectations on the right hand side. Finding the fixed point solution of these ODEs then gives an approximation of the desired steady state mean rates of increase of the rewards.

We also note that the Theorem 4.1 implies that the normalised accumulated rewards are deterministic in the steady state limit and therefore all the higher moments are just products of the respective expectations. Specifically, the variance of these measures is zero.

4.2 Transient rewards

We show how to derive approximations for means of the accumulated rewards at each time t, the quantities ($\overline{\mathrm{E}}$). For our sample rewards, these are

$$\mathbb{E}[A_P(t)] \text{ and } \mathbb{E}[A_{PR}(t)] \text{ for } t \geq 0$$

Such expectations are differentiable in general, so we can set a new differential equation for the mean of each accumulated component count. To obtain the right hand side, we note that since the rewards are always bounded and differentiable, we can swap the differentiation and expectation to get

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\left[\int_0^t X(u)\mathrm{d}u\right] = \mathbb{E}[X(t)] \tag{4.4}$$

This can be generalised to the case when $X(u)$ is replaced by a general product $\prod_{i=1}^n X_i^{r_i}(u)$. The right hand side is then always one of the moments, for which there is an approximation by the solution to one of the ODEs. Numerically solving these simultaneously gives an approximation to the means of accumulated rewards at each time point t. For our sample rewards we would take ODEs approximating all the moments up to order two (so that they include the joint moment $\mathbb{E}[P_0(t)R_0(t)]$) and add the following ODEs to the system:

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}\left[\int_0^t P_i(u)\mathrm{d}u\right] = \widetilde{\mathbb{E}}[P_i(t)],\ i=0,1 \tag{4.5}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}\left[\int_0^t P_0(u)R_0(u)\mathrm{d}u\right] = \widetilde{\mathbb{E}}[P_0(t)R_0(t)] \tag{4.6}$$

Numerically solving these gives approximations to the expectations of the rewards

$$\begin{aligned}\mathbb{E}[A_P(t)] &\approx c_0\widetilde{\mathbb{E}}\left[\int_0^t P_0(u)\mathrm{d}u\right] + c_1\widetilde{\mathbb{E}}\left[\int_0^t P_1(u)\mathrm{d}u\right]\\ \mathbb{E}\left[A_{PR}(t)\right] &\approx c\widetilde{\mathbb{E}}\left[\int_0^t P_0(u)R_0(u)\mathrm{d}u\right]\end{aligned}$$

4.3 Higher moments of rewards

We now look at the general case of higher moments of the accumulated component counts at each time t, the quantities ($\overline{\mathrm{M}}$). Using these, we can then get for example approximations to the variances of accumulated rewards, such as:

$$\mathrm{Var}[A_P(t)] \text{ and } \mathrm{Var}[A_{PR}(t)] \text{ for } t \geq 0$$

First, define the shorthand for the accumulated count of X up to time t as:

$$\overline{X}(t) = \int_0^t X(u)\mathrm{d}u \tag{4.7}$$

For simplicity, we first look at the second order moments of accumulated counts and later show how to extend the technique to higher orders. As for the case of mean component counts, we can note that the moments are differentiable and bounded and so we can swap the differentiation and expectation and get ODEs of the form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\left[\overline{X}(t)\overline{Y}(t)\right] = \mathbb{E}[X(t)\overline{Y}(t)] + \mathbb{E}[Y(t)\overline{X}(t)] \tag{4.8}$$

The right hand side now contains expectations of the form $\mathbb{E}[X(t)\overline{Y}(t)]$. We can try to define ODEs governing these. This time, the function $X(t)\overline{Y}(t)$ is not differentiable, so we cannot simply swap the expectation and differentiation. We can look at the derivative of the expectation $\mathbb{E}[X(t)\overline{Y}(t)]$ from the first principles to arrive at the following theorem (with full proof in Appendix A):

THEOREM 4.2.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[X(t)\overline{Y}(t)] = \mathbb{E}[f_X(t)\overline{Y}(t)] + \mathbb{E}[X(t)Y(t)] \tag{4.9}$$

where $f_X(t)$ involves only component counts (and no expectations) such that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\left[X(t)\right] = \mathbb{E}[f_X(t)] \tag{4.10}$$

To be able to solve this ODE together with the rest of the system numerically, we need to be able to evaluate (or approximate) both the summands in the right hand side. The second term $\mathbb{E}[X(t)Y(t)]$ has an approximation $\widetilde{\mathbb{E}}[X(t)Y(t)]$ given by one of the moment ODEs. If the model is split-free, the first term contains, after moving the expectation through

summations and multiplications by constants, terms of the form:

$$\mathbb{E}[Z(t)\overline{Y}(t)] \text{ and } \mathbb{E}[\min(g(t)\overline{Y}(t), h(t)\overline{Y}(t))]$$

where g, h are piecewise linear functions (i.e. involve only linear combinations and applications of the min function) of the component counts. The Theorem 4.2 can be repeatedly used to obtain ODEs of the former terms. For the latter ones, we can apply the approximation of Equation (3.2) repeatedly to get piecewise linear functions involving the terms $\mathbb{E}[Z(t)\overline{Y}(t)]$.
If the model is splitting, we additionally get terms:

$$\mathbb{E}[f(X_1(t), \ldots, X_m(t), X_{m+1}(t)\overline{Y}(t), \ldots, X_k(t)\overline{Y}(t))]$$

where f is a rational function of the component counts. We can then apply the approximation of Equation (3.3) to get rational functions involving the terms $\mathbb{E}[Z(t)\overline{Y}(t)]$.
We demonstrate this to find the variance of $A_P(t)$. We have:

$$\begin{aligned}\mathrm{Var}[c_0\overline{P_0}(t) + c_1\overline{P_1}(t)] = c_0^2\mathrm{Var}[\overline{P_0}(t)] + c_1^2\mathrm{Var}[\overline{P_1}(t)] \\ + 2c_1c_2\mathrm{Cov}[\overline{P_0}(t), \overline{P_1}(t)]\end{aligned}$$

and also:

$$\begin{aligned}\mathrm{Var}[\overline{P_i}(t)] &= \mathbb{E}[(\overline{P_i}(t))^2] - \mathbb{E}[\overline{P_i}(t)]^2 \\ \mathrm{Cov}[\overline{P_0}(t), \overline{P_1}(t)] &= \mathbb{E}[\overline{P_0}(t)\overline{P_1}(t)] - \mathbb{E}[\overline{P_0}(t)]\mathbb{E}[\overline{P_1}(t)]\end{aligned}$$

Therefore, in addition to the ODEs from Equation (4.5), we need:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}[(\overline{P_i}(t))^2] &= 2\widetilde{\mathbb{E}}[P_i(t)\overline{P_i}(t)] \\ \frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}[\overline{P_0}(t)\overline{P_1}(t)] &= \widetilde{\mathbb{E}}[P_0(t)\overline{P_1}(t)] + \widetilde{\mathbb{E}}[P_1(t)\overline{P_0}(t)]\end{aligned}$$

To get the ODEs for $\widetilde{\mathbb{E}}[P_0(t)\overline{P_i}(t)]$ for $i = 0, 1$, we can use Theorem 4.2 where:

$$f_{P_0}(t) = qP_1(t) - \min(r_1P_0(t), r_2R_0(t))$$

Therefore

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}\left[P_0(t)\overline{P_i}(t)\right] = q\widetilde{\mathbb{E}}[P_1(t)\overline{P_i}(t)] + \widetilde{\mathbb{E}}[P_0(t)P_i(t)] \\ - \min\left(r_1\widetilde{\mathbb{E}}\left[P_0(t)\overline{P_i}(t)\right], r_2\widetilde{\mathbb{E}}\left[R_0(t)\overline{P_i}(t)\right]\right)\end{aligned}$$

Similarly we can obtain ODEs for the other terms $\widetilde{\mathbb{E}}[P_i(t)\overline{P_j}(t)]$ and $\widetilde{\mathbb{E}}[R_i(t)\overline{P_j}(t)]$. To complete the system of ODEs so that all the moment terms on the right hand sides have a corresponding differential equation, we add ODEs for the second moments such as $\widetilde{\mathbb{E}}[P_0(t)P_1(t)]$.
Numerically solving the resulting system of ODEs then gives us the approximation to the variance of $A_P(t)$.
Theorem 4.2 can be generalised to cover the case when X and Y are *products* of component counts. We can extend the notation for accumulated counts to accumulated products of counts: if $M(t) = \prod_i X_i^{k_i}(t)$, we let:

$$\overline{M}(t) = \int_0^t \prod_i X_i^{k_i}(u)\mathrm{d}u \tag{4.11}$$

For example, to get the variance of A_{PR} we get:

$$\mathrm{Var}[c\overline{P_0R_0}(t)] = c^2\mathbb{E}[(\overline{P_0R_0}(t))^2] - c^2\mathbb{E}[\overline{P_0R_0}(t)]^2$$

This requires the ODEs:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}[\overline{P_0R_0}(t)^2] &= 2\widetilde{\mathbb{E}}[P_0(t)R_0(t)\overline{P_0R_0}(t)] \\ \frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbb{E}}[P_0(t)R_0(t)\overline{P_0R_0}(t)] &= \widetilde{\mathbb{E}}[f_{P_0R_0}(t)\overline{P_0R_0}(t)] \\ &\quad + \widetilde{\mathbb{E}}[P_0(t)^2R_0(t)^2]\end{aligned}$$

We can expand the term $\widetilde{\mathbb{E}}[f_{P_0R_0}(t)\overline{P_0R_0}(t)]$ using Equation (3.1) (more precisely its form before applying the approximations) to get terms such as $\mathbb{E}[P_1(t)\overline{P_0R_0}(t)]$. Taking ODEs for these and also for the moments up to the order 4, we can solve the resulting system numerically and get an approximation to the variance of $A_{PR}(t)$.
Furthermore, we can easily extend the Theorem 4.2 to cover arbitrary moments of accumulated products:

$$\mathbb{E}\left[M_0(t)^{k_0}\overline{M_1}(t)^{k_1}\cdots\overline{M_l}(t)^{k_l}\right] \tag{4.12}$$

For the general statement see Theorem A.1 in Appendix A.

4.4 Convergence

As presented in more detail in [6], using a result of Kurtz [18] it can be shown that the first moments of component counts converge to their ODE approximation when scaled by the total component population size, as the component populations are scaled up. Let $X_n(t)$ be the stochastic process corresponding to the count of X components in a GPEPA model where the total component population has been scaled up by $n \in \mathbb{Z}_+$, then the result is that $\frac{1}{n}|\mathbb{E}[X_n(t)] - \widetilde{\mathbb{E}}[X_n(t)]| \to 0$ as $n \to \infty$. Applying Fubini's theorem and dominated convergence, it is straightforward to show that the first moments of linear accumulated rewards converge similarly, that is, $\frac{1}{n}|\mathbb{E}[\overline{X_n}(t)] - \widetilde{\mathbb{E}}[\overline{X_n}(t)]| \to 0$ as $n \to \infty$.
An argument was also presented in [6] suggesting that we would expect similar convergence of covariances of component counts to their ODE approximations, for example, $\frac{1}{n}|\widetilde{\mathrm{Var}}[X_n(t)] - \mathrm{Var}[X_n(t)]| \to 0$ as $n \to \infty$. Although we do not have space to give the argument here, we believe that an analogous result can be shown for covariances of linear accumulated rewards. That is, we would expect for example that $\frac{1}{n}|\widetilde{\mathrm{Var}}[\overline{X_n}(t)] - \mathrm{Var}[\overline{X_n}(t)]| \to 0$ and that $\frac{1}{n}|\widetilde{\mathrm{Cov}}[\overline{X_n}(t), X_n(t)] - \mathrm{Cov}[\overline{X_n}(t), X_n(t)]| \to 0$, as $n \to \infty$.

4.5 Complexity

The total complexity of the presented technique to obtain moments of accumulated rewards is given by the number of ODEs and the time over which they need to be numerically integrated.
Similar to the original method for moments of component counts, it is necessary to include ODEs of all moments up to a certain order. More specifically, if there are C different components, the method gives $O(C^n)$ ODEs corresponding to all moments up to order n.
For example, in the processors/resources model $C = 4$ and to get the mean of $A_P(t)$, the method requires 6 ODEs. For the variance (the order is 2), there are 27 ODEs. In case of the mean of A_{PR}, 5 ODEs are needed and for the variance (the order is 4 for the joint moment), 85 ODEs have to be solved.
It is worth noting that the usual algorithms for numerically solving systems of ODEs have both run time and memory requirements linearly dependent on the size of the system.

Therefore the technique is able to cope with fairly large systems. For example models requiring more than 10^4 ODEs can be solved in under a minute on a standard Core 2 Duo 3.0 GHz desktop computer.

5. EXAMPLE

We demonstrate the presented techniques on an example of a massively parallel system as mentioned in the introduction. The model consists of a large number of clients and servers cooperating together. The clients use the servers in two stages – first request some data and then obtain the data, and then perform a task individually. The servers, in addition to serving clients, can hibernate to save energy and can also break. Broken servers need to be repaired. The full Grouped PEPA description of this model is:

$$\begin{aligned}
Client &\stackrel{def}{=} (request, r_{req}).Client_{requested} \\
&\quad + (wait, r_{wait}).Client_{waiting} \\
Client_{requested} &\stackrel{def}{=} (data, r_{data}).Client_{data} \\
Client_{data} &\stackrel{def}{=} (task, r_{task}).Client \\
Client_{waiting} &\stackrel{def}{=} (resume, r_{res}).Client \\
\\
Server &\stackrel{def}{=} (request, r_{res}).Server_{requested} \\
&\quad + (sleep, r_{sleep}).Server_{sleep} \\
&\quad + (break, r_{break}).Server_{broken} \\
Server_{requested} &\stackrel{def}{=} (data, r_{data}).Server_{data} \\
Server_{data} &\stackrel{def}{=} (reset, r_{reset}).Server \\
Server_{sleep} &\stackrel{def}{=} (wakeup, r_{wakeup}).Server \\
Server_{broken} &\stackrel{def}{=} (repair, r_{repair}).Server_{sleep}
\end{aligned}$$

$$\mathbf{Servers}\{Server[s]\} \underset{\{request, data\}}{\bowtie} \mathbf{Clients}\{Client[c]\}$$

To get an initial idea of how the system behaves over time, we look at the counts of the individual states of the client and server components. Figure 3 shows these for the system with initial number of clients, $c = 200$, and servers, $s = 20$ and rates with values given in Appendix B.

5.1 Accumulated rewards

We can define simple rewards on this model. To model power consumption of the servers, we assume that servers consume energy at a rate specific to each state during the time of being in that state. The total reward for power consumption is then the accumulation of the corresponding linear combination of the individual state counts:

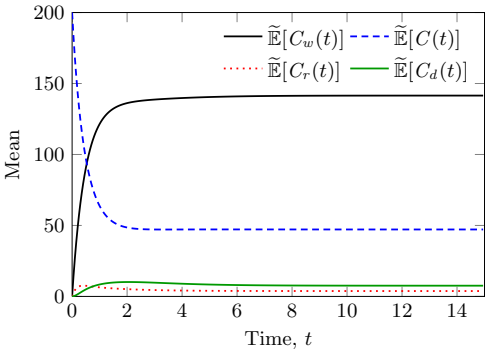
$$\begin{aligned}
A_{\text{power}}(t) &= c_{\text{running}}\overline{S_r}(t) + c_{\text{power}}\overline{S}(t) \\
&\quad + c_{\text{broken}}\overline{S_b}(t) + c_{\text{data}}\overline{S_d}(t)
\end{aligned} \tag{5.1}$$

Similarly, income from serving the clients can be modelled by accumulating at a certain rate in the state of the server fulfilling a request:

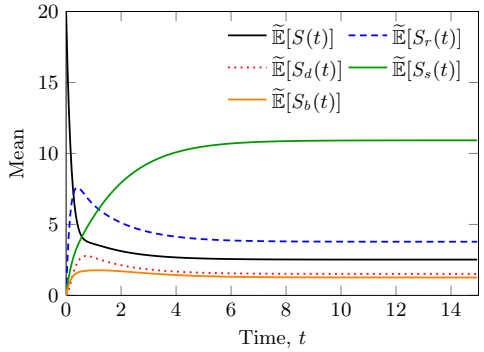
$$A_{\text{income}}(t) = c_{\text{fee}}\overline{S_r}(t) \tag{5.2}$$

Finally, we can look at the hypothetical total income that could result from the difference between the income from serving the clients and the energy cost:

$$A_{\text{total}}(t) = A_{\text{income}}(t) - A_{\text{power}}(t) \tag{5.3}$$



(a) Clients



(b) Servers

Figure 3: Means of component counts in the client–server model.

Figure 4 shows the evolution of the mean of the reward $A_{\text{total}}(t)$ over time, with the accumulation rates given in Appendix B. It contains the ODE approximation of the transient mean and also indicates the variability of the reward by showing its standard deviation. The accuracy of the approximation is shown by plotting the difference from the exact quantities obtained via simulation. We can see that the error is in the order of 0.1% of the value of the reward.

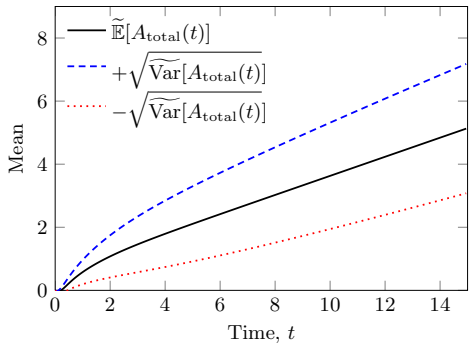
5.2 Completion times

We illustrate how to use the moments of accumulated component counts to obtain approximation to the completion time measures of the model, quantity $(\overline{\text{P}})$. Consider the random variable representing the first time the general reward $A(t)$ hits a target value a:

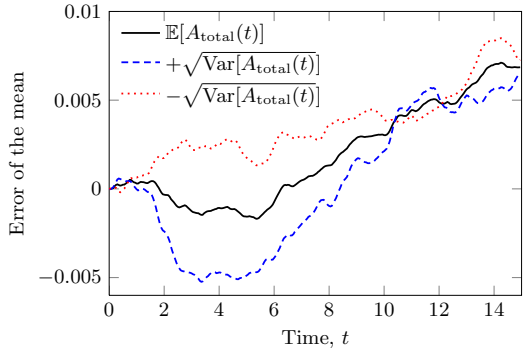
$$C = \inf\{t \geq 0 : A(t) > a\} \tag{5.4}$$

In order to guarantee various service level agreements of the form "the probability of reaching a reward a in time t is less than p", we are interested in the distribution of C, i.e. in the probabilities $\mathbb{P}(C \leq t)$. In [7], the authors addressed the same problem for the case of completion times of component counts. They used the well known one sided improvement of Chebyshev's inequality: for a random variable X:

$$\begin{aligned}
\mathbb{P}(X - \mathbb{E}[X] \geq y) &\leq \frac{\text{Var}[X]}{\text{Var}[X] + y^2} \\
\mathbb{P}(\mathbb{E}[X] - X \geq y) &\leq \frac{\text{Var}[X]}{\text{Var}[X] + y^2}
\end{aligned} \tag{5.5}$$

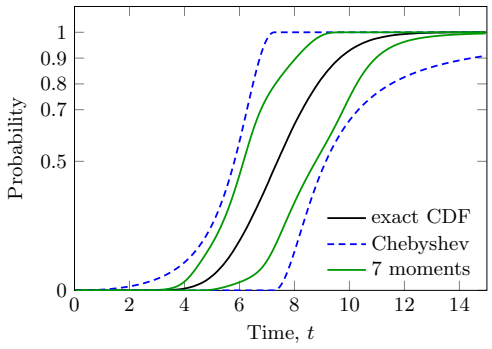


(a) Mean reward

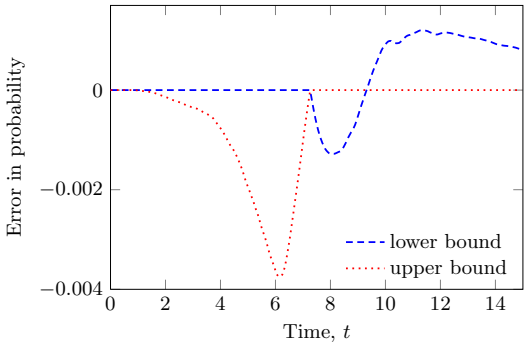


(b) Error of order 10^{-2} in the mean reward

Figure 4: Approximation of the total reward.



(a) Lower and upper approximation of the completion time CDF



(b) Absolute error of order 10^{-3} in CDF approximations based on the Chebyshev's inequality

Figure 5: Approximations of the CDF of the time of the energy cost reward reaching $a = 2.0$.

To get the required probabilities, we note that if $r(t)$ is non-decreasing:

$$\mathbb{P}(C \leq t) = \mathbb{P}(A(t) \geq a)$$

This allows us to use the following bounds:

$$\mathbb{P}(C \leq t) \leq \frac{\text{Var}[r(t)]}{\text{Var}[A(t)] + (\mathbb{E}[A(t)] - a)^2} \qquad \text{if } \mathbb{E}[A(t)] \leq a$$
$$\mathbb{P}(C \leq t) \geq 1 - \frac{\text{Var}[A(t)]}{\text{Var}[A(t)] + (\mathbb{E}[A(t)] - a)^2} \qquad \text{if } \mathbb{E}[A(t)] > a \tag{5.6}$$

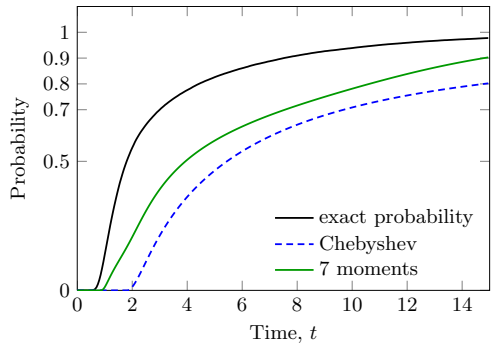
If $A(t)$ can decrease, we have instead:

$$\mathbb{P}(C \leq t) \geq \mathbb{P}(A(t) \geq a)$$

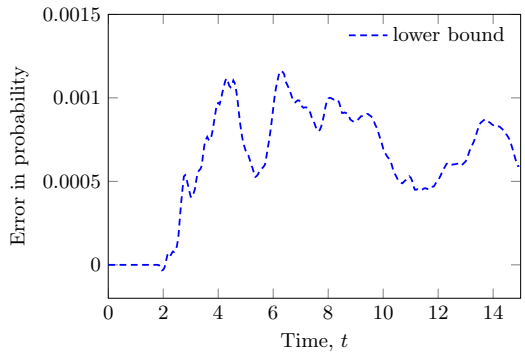
and only the lower bound in Equation (5.6) can be used.
The right hand sides of both these bounds can be quickly approximated by solution to the systems of ODEs described earlier in this paper, thus giving a lower and upper approximations to the CDF of completion times.
Figure 5 shows the lower and upper approximations to the CDF of the completion time of the reward A_{power} reaching target $a = 2.0$ and compares them to the CDF estimated from simulation. As the rate of this reward is non-negative at each time, both the upper and lower approximations can be used. The corresponding error, also shown in the figure, is in the order of 0.1% of the probability.
In case of the total reward A_{total}, only the lower approximation can be used, since the rate of accumulation can be negative. Figure 6 shows the approximated lower approximation for the total reward and the corresponding error, which is of the order 0.1% of the probability.
Comparing with the CDF, we can see that the bounds are quite loose. However, the lower bounds can still give a useful conservative estimate of the completion times. Moreover, the ODE approximations of moments can be readily used to produce more precise bounds based on moments of higher orders than the variance. Figures 5(a) and 6(a) show CDF approximations obtained from the first 7 moments using the method in [19]. The error in these is of the same order, 10^{-3}, as the error of the approximations based on the Chebyshev's inequality.

5.3 Approximations

While the original ODEs for the moments of component counts and the presented extension to the moments of rewards often give accurate results, larger errors can occur. In [6] the authors investigate the dynamics of the error of the approximation in the component counts. It has been shown that the largest error occurs around so-called *switch points*. These only occur for certain parameter regimes which lead to the approximation Equation (3.2) being at its worst – when the two arguments to the min function are close – and represent the situation when the model switches between two different modes of behaviour. It was proposed that plot-

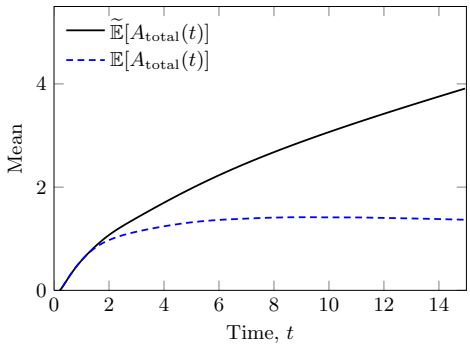


(a) Lower and upper approximation of the completion time CDF

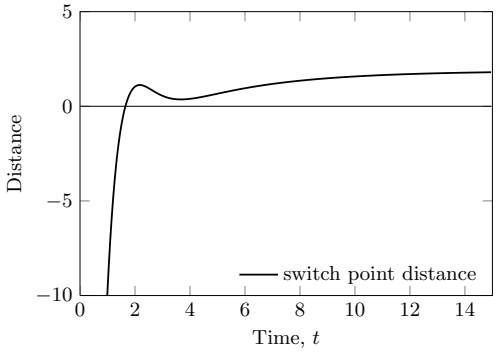


(b) Absolute error of order 10^{-3} in CDF approximations based on the Chebyshev's inequality

Figure 6: Approximations of the CDF of the time of the total reward reaching $R = 1.0$.



(a) Mean of $A_{\text{total}}(t)$



(b) Distance from a switch point

Figure 7: Example of an ODE approximation of mean reward with high error.

ting the distance from a switch point can serve as a rough indicator of the accuracy of the model.

The same reasoning applies to the case of moments of accumulated rewards. The difference is that the error can get amplified by accumulation over time and also by constant multiplication in the linear combinations specifying rewards, such as c_{power}. For example, Figure 7 shows an instance when certain rate parameters cause the error to be particularly visible.

Figure 7(a) shows both quantitative and qualitative differences in the approximation of the mean of $A_{\text{total}}(t)$. Figure 7(b) explains this by showing the distance the model maintains from being near a switch point. It shows that around the time $t \approx 2.0$, the model approaches a switch point. This coincides with the time when the ODE approximation to the mean of the reward starts to differ from the exact value. It is worth noting that the error in the mean component counts is still only about 1%. Also, supporting the arguments from Section 4.4, the relative error disappears as the system size scales up.

5.4 Trade-off between energy consumption and performance

A key application of the efficient reward computation techniques presented in this paper is the design of systems with *both* performance and energy consumption requirements. In the client–server model, we might be interested in the optimal number of servers that have to be employed in order to guarantee given performance requirements while minimising the associated running costs. The performance requirements are often given in terms of a *service level agreement* (SLA) for each client. In the context of this model, a suitable SLA might require that a client finishes its first *think* action within a given time period with a given high probability, for example within time $t = 4.0$ with probability at least 0.9. Considering only the configurations that satisfy such an SLA, the *feasible configurations*, we can look for those that minimise certain cost function, such as the mean of $A_{\text{power}}(t)$ from Section 5.1.

Figure 8 shows an example where we vary the number of servers and the rate with which they are put to sleep. For each configuration we calculate the energy used and plot a point on the surface only if that configuration satisfies the SLA requirement mentioned above. We are able to find a configuration (84 servers and a hibernation rate of 0.37) which minimises the energy consumption in the system.

Intuitively, increasing the number of servers and decreasing the hibernation rate increases the probability of a client finishing early, but also raises the energy cost of running the system. The passage time probabilities can be accessed using the component count ODEs as shown in [7] and the mean running cost $A_{\text{power}}(t)$ can be accessed from the accumulated reward ODEs as shown in this paper. Numerically solving the resulting system of ODEs is computationally cheap and

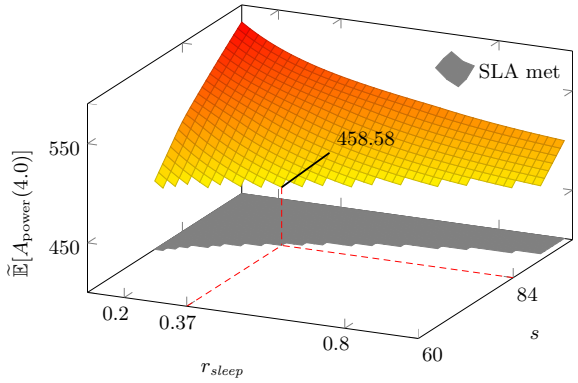


Figure 8: Global optimisation of the energy consumption of the server components. Only configurations satisfying the SLA that each individual client finishes its first *think* action within 4.0 seconds with probability at least 0.9 are shown.

can be therefore repeated for a large number of parameter combinations. The plot in Figure 8 was produced by solving the system of ODEs for 1600 different parameter combinations taking a few seconds on a standard desktop computer. Moreover, although not guaranteed to find the global minimum in general, well-known local optimisation techniques, such as the *interior point* algorithm can be directly applied. In this case, we were able to find a local minimum of cost $A_{\text{power}} = 457.71$ at $r_{sleep} = 0.34$, $s = 83.22$ using the MATLAB *fmincon* implementation of the interior point algorithm. This required ODE solutions for only around 50 distinct parameter combinations. The reason for the difference in the two minima (between MATLAB local optimisation and parameter sweeping) is that the local optimisation algorithm does not exclude non-integer valued server counts. Therefore in order to obtain a meaningful configuration we can round the number of servers up to $s = 84$. This can easily be verified as still feasible and it achieves a cost of 461.98, which is still very close to the minimum found by exploring the large parameter space above.

6. CONCLUSION AND FUTURE WORK

In this paper, we have shown how energy consumption and other accumulated measures can be extracted from massively parallel stochastic systems. To achieve this, we have embedded the appropriate Markov reward structures in sets of differential equations which can be integrated numerically. This gives access to solution to reward problems based on substantially larger Markov models than has previously been possible in for example [8], which analysed Markov Reward Models of order 10^6 states. This contrasts with the reward model which, by the nature of differential equation analysis, we are able to analyse in this paper which has approximately 10^{134} explicit states or 10^{12} states in its aggregate form.

Since the solution of these systems of ODEs is so rapid, we have been able to make use of these reward techniques to solve a sensitivity analysis problem. We have been able to combine an SLA requirement, in the form of passage-time quantile, with the computation of a reward structure. Figure 8 showed which feasible model parameters satisfied the SLA and how much energy would be expended doing so. We were able to use optimisation techniques to find a minimum energy expenditure for the model. Similar techniques can be used to obtain model parameters, such as the rates in Appendix B for the client–server model, when fitting models to measurement data from real systems. There is potential for more efficient optimisation algorithms to take advantage of the ODE form of the cost function. In future, we plan to adapt such techniques, for example those in [20, 21] guaranteeing global optima, to the case of the systems of ODEs arising from large scale Markov models.

While massive state space Markov reward analysis is now feasible, one of the recognised features of differential equation analysis is that it is an approximation of the explicit state space model [5]. In most cases, the agreement between differential equation-based reward analysis and simulation is very good, as for instance in Figure 4. However, where model parameters give rise to so-called switch point behaviour, error can accumulate for these very particular parameter regimes [6]. Fortunately, these less accurate cases can be predicted, avoided (by parameter modification) or potentially simulated instead.

As with previous work [11], higher moments can be generated for reward measures but, again, the size of the model that can be analysed is the novel feature. Second moments, in particular, give a very straight-forward indication of precision of the first order solution, and we have shown this in Figure 4.

The completion time until a reward level is reached is an extremely useful reward metric and, using higher reward moments, upper and lower bounds on the completion-time distribution can be derived. The distribution lower bound, in particular, is useful for generating conservative quantile measurements which satisfy industrial service level agreements. The CDF approximations based on Chebyshev's inequality in Figures 5 and 6 are coarse in this paper, due to using only the first two moments. However, as the same figures show, it is possible to get stronger constraints still using the first 7 moments. We plan to investigate, again using the techniques from Tari *et al.* [19], the production of bounds for up to the first 20 moments. We aim to improve the efficiency of the presented method to produce these moments of high order as well as investigate the possibility of bounds based on joint moments.

Additionally, we will look at incorporating extra types of reward measures into the ODE analysis method, for instance impulse rewards [3] and partial reward models [9].

References

[1] R. M. Smith, K. S. Trivedi, and A. V. Ramesh, "Performability analysis: Measures, an algorithm, and a case study," *IEEE Transactions on Computers*, vol. 37, pp. 406–417, Apr. 1988.

[2] L. Donatiello and V. Grassi, "On evaluating the cumulative performance distribution of fault-tolerant computer systems," *IEEE Transactions on Computers*, vol. 40, pp. 1301–1307, Nov. 1991.

[3] E. de Souza e Silva and R. Gail, "An algorithm to calculate transient distributions of cummulative rate and impulse-based rewards," *Communications in Statistics: Stochastic Models*, vol. 14, no. 3, pp. 509–536, 1998.

[4] J. Hillston, "Fluid flow approximation of PEPA models," in *QEST'05, Proceedings of the 2nd International Conference on Quantitative Evaluation of Sys-*

tems, (Torino), pp. 33–42, IEEE Computer Society Press, September 2005.
[5] R. Hayden and J. T. Bradley, "A fluid analysis framework for a Markovian process algebra," *Theoretical Computer Science*, vol. 411, pp. 2260–2297, May 2010.
[6] A. Stefanek, R. Hayden, and J. T. Bradley, "A new tool for the performance analysis of massively parallel computer systems," in *Eighth Workshop on Quantitative Aspects of Programming Languages (QAPL 2010)*, Electronic Proceedings in Theoretical Computer Science, March 2010.
[7] R. Hayden, A. Stefanek, and J. T. Bradley, "Fluid computation of passage time distributions in large Markov models." Submitted for publication, Nov. 2010.
[8] M. Telek and S. Rácz, "Numerical analysis of large Markovian reward models," *Performance Evaluation*, vol. 36–37, pp. 95–114, Aug. 1999.
[9] A. Bobbio, V. G. Kulkarni, and M. Telek, "Partial loss in reward models," in *MMR'00, 2nd International Conference on Mathematical Methods in Reliability*, (Bordeaux), pp. 207–210, July 2000.
[10] G. Horváth, S. Rácz, A. Tari, and M. Telek, "Evaluation of reward analysis methods with MRMSolve 2.0," *QEST'04, International Conference on Quantitative Evaluation of Systems*, pp. 165–174, 2004.
[11] M. Telek, A. Horváth, and G. Horváth, "Analysis of inhomogeneous Markov reward models," *Linear Algebra and its Applications*, vol. 386, pp. 383–405, 2004.
[12] H. Nabli and B. Sericola, "Performability analysis: A new algorithm," *IEEE Transactions on Computers*, vol. 45, pp. 491–494, Apr. 1996.
[13] H. C. Tijms and R. Veldman, "A fast algorithm for the transient reward distribution in continuous-time Markov chains," *Operations Research Letters*, vol. 26, no. 4, pp. 155–158, 2000.
[14] F. Castella, G. Dujardin, and B. Sericola, "Moments analysis in homogeneous Markov reward models," *Methodology and Computing in Applied Probability*, vol. 11, pp. 583–601, 2009.
[15] D. T. Gillespie, *Markov Processes: An Introduction for Physical Scientists*. Academic Press, 1991.
[16] J. Ding, *Structural and Fluid Analysis for Large Scale PEPA Models–With Applications to Content Adaptation Systems*. PhD thesis, The University of Edinburgh, 2009.
[17] J. R. Norris, *Markov chains*. Cambridge University Press, 1998.
[18] T. Kurtz, "Solutions of ordinary differential equations as limits of pure jump Markov processes," *Journal of Applied Probability*, vol. 7, pp. 49–58, April 1970.
[19] A. Tari, M. Telek, and P. Buchholz, "A unified approach to the moments-based distribution estimation–Unbounded support," in *EPEW'05, European Performance Engineering Workshop*, vol. 3670 of *Lecture Notes in Computer Science*, (Versailles), pp. 79–93, Springer, 2005.
[20] I. Papamichail and C. S. Adjiman, "A Rigorous Global Optimization Algorithm for Problems with Ordinary Differential Equations," *Journal of Global Optimization*, no. 1992, pp. 1–33, 2002.
[21] A. B. Singer and P. I. Barton, "Global Optimization with Nonlinear Ordinary Differential Equations," *Journal of Global Optimization*, vol. 34, pp. 159–190, Feb. 2006.

APPENDIX

A. PROOFS

PROOF OF THEOREM 4.2. We have

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[X(t)\overline{Y}(t)]\\
&= \lim_{h\to 0}\frac{1}{h}(\mathbb{E}[X(t+h)\overline{Y}(t+h)] - \mathbb{E}[X(t)\overline{Y}(t)])\\
&= \lim_{h\to 0}\frac{1}{h}(\mathbb{E}[(X(t+h)-X(t))\overline{Y}(t)]\\
&\qquad + \mathbb{E}[X(t+h)(\overline{Y}(t+h)-\overline{Y}(t))])\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[X(t)\overline{Y}(s)]|_{s=t} + \mathbb{E}[X(t)Y(t)]
\end{aligned}$$

where the second term is obtained by applying the bounded convergence theorem. For the first term

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[X(t)\overline{Y}(s)]\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\int_i\sum_k ik\mathbb{P}(X(t)=k,\overline{Y}(s)=i)\mathrm{d}i\\
&= \int_i\sum_k ik\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{P}(X(t)=k \mid \overline{Y}(s)=i)\,\mathbb{P}(\overline{Y}(s)=i)\mathrm{d}i\\
&= \int_i i\mathbb{E}[f_X(t) \mid \overline{Y}(s)=i]\,\mathbb{P}(\overline{Y}(s)=i)\\
&= \mathbb{E}[f_X(t)\overline{Y}(s)]
\end{aligned}$$

and the result follows. □

THEOREM A.1 (THE GENERAL FORM OF THEOREM 4.2). *For finite products M_i, $i=0,\ldots,N$ of the form*

$$M_i(t) = \prod_h X_h(t)^{k_{ih}} \qquad k_{ih}\geq 0$$

and $m_j > 0$ for $j=1,\ldots,N$ we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\left[M_0(t)\prod_{j=1}^N \overline{M_j}(t)^{m_j}\right] &= \mathbb{E}\left[f_{M_0}(t)\prod_{j=1}^N \overline{M_j}(t)^{m_j}\right]\\
&\quad + \sum_{n=1}^N m_n\mathbb{E}\left[M'_n(t)\prod_{j=1}^N \overline{M_j}(t)^{m_j-[n=j]}\right]
\end{aligned}$$

where

$$M'_n(t) = \prod_h X_h(t)^{k_{0h}+k_{nh}}$$

and f_{M_0} is a function involving products of component counts and no expectations such that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[M_0(t)] = \mathbb{E}[f_{M_0}(t)]$$

and $[n=j]=1$ if $n=j$ and 0 otherwise.

B. RATES

Rates used in the client–server model:

$r_{req}=3$	$r_{wait}=1.5$	$r_{data}=2$	$r_{task}=1$	$r_{res}=0.5$
$r_{reset}=5$	$r_{sleep}=0.8$	$r_{wakeup}=0.4$	$r_{break}=0.5$	$r_{repair}=1$

Accumulation rates used in A_{power} and A_{income}:

$c_{\text{power}}=0.05$	$c_{\text{running}}=0.1$	$c_{\text{broken}}=0.2$
$c_{\text{data}}=0.05$	$c_{\text{fee}}=0.2$	