Mean-field approximations for performance models with generally-timed transitions

Richard A. Hayden
Department of Computing
Imperial College London, UK
rh@doc.ic.ac.uk

ABSTRACT

We show how the popular mean-field approach for analysing Markovian models of massively-parallel computer systems can be extended to incorporate generally-timed transitions. Specifically, in the context of a simple peer-to-peer software update model, we show how systems of delay differential equations can be formally derived from such models. Finally, we verify empirically that the approximation is accurate and converges under the usual mean-field scaling.

1. INTRODUCTION

Mean-field or *fluid-approximation* techniques have become very popular recently as a means of addressing the state-space explosion problem for Markov chains consisting of large populations of interacting components [3, 4, 2, 7]. This kind of approach is usually based on the construction of ordinary differential equations (ODEs) from a Markovian model which approximate the evolution of the mean number of components in each of their local states over time.

In this paper, by means of a simple example, we will show how this style of approach can be extended to deal with performance models allowing both exponentially- and deterministically-timed transitions. In particular, we will see how this allows us to analyse large systems incorporating deterministic timeouts, which are an important feature of many real-world computer or networking protocols. Finally, we will show how the approach can be extended further to allow generally-timed transitions.

The underlying discrete stochastic process of such a model is a *generalised semi-Markov process (GSMP)* [10]. In particular, the process is no longer necessarily Markovian since the future evolution of the model may depend on the amount of time elapsed for any enabled non-exponentially-timed transition. Although general theoretical techniques exist for the analysis of such GSMP models (e.g. [6, 8]), it is not surprising that they suffer from the state-space explosion problem just as in the purely Markovian case. Indeed, the complexity of these techniques still depends at least linearly on the number of discrete states in the model and thus exponentially on the number of interacting entities. Furthermore, many of these approaches impose very significant structural requirements on the enabling of non-exponential transitions (e.g. [11]). An alternative approach would be to approximate all non-exponential times by suitable phase-type distributions. For example, deterministic times can be approximated by k-stage Erlang distributions. Replacing generally-distributed holding times with their phase-type approximations results in an approximate Markovian model which could then be treated with the existing Markovian mean-field approaches. However, this is often impractical since the number of phases must often be very large to obtain an accurate approximation.

Mirroring the Markovian mean-field approach, we will show instead how a system of *delay differential equations (DDEs)*[1] approximating mean component counts can be derived directly from the dynamics of the underlying stochastic process. This small system of equations can then be solved numerically in a scalable manner since its size is independent of those of the component populations.

2. A SIMPLE EXAMPLE

We consider a very simple model of a peer-to-peer software update process. There are two general classes of nodes in this system which we term *old* and *updated*. Old nodes are those running the old software version and new nodes are those which have been updated to the new version. Both types of nodes alternate between being *on* and *off*. When an updated node is on, an old node may locate it and subsequently update itself in a peer-to-peer fashion. Whenever an old node comes on, it polls the network for new nodes (so it can be updated) before *giving up if it does not find one in a deterministic amount of time* γ. Updated nodes have two states which are just on and off, which we write as u and v, respectively. Old nodes have three states: on (y), off (x) and a state representing an old node which is on but has given up seeking updates (z). All transitions except for the deterministic timeout are Markovian and are given in Table 1, where N is the total component population. We follow the usual methodology for the stochastic modelling

Transition	Rate
$x \to y$	λ
$y \to x$	ρ
$z \to x$	ρ
$y \to u$	$\frac{\beta \mathbf{u}}{N}$
$v \to u$	λ
$u \to v$	ρ

Table 1: Markovian transitions for peer-to-peer software update model, where $\mathbf{u}$ is the current number of nodes in the state u.

[1]As opposed to ODEs, DDEs specify derivatives of quantities in terms of both their current and their historical values.

.

of epidemics in that the chance of an old node finding an updated one in a small period of time is proportional to the number of available updated nodes.

2.1 Derivation of DDEs

Let $\mathbf{x}_t$ be the number of nodes in state x at time t and similarly for the other states. Then $\boldsymbol{S}_t = (\mathbf{x}_t, \mathbf{y}_t, \mathbf{z}_t, \mathbf{u}_t, \mathbf{v}_t)$ is the discrete aggregated state vector. Also individual states of this process are written $\boldsymbol{s} = (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$ and furthermore, the notation $\boldsymbol{s}_{\mathbf{y}^{+1},\mathbf{u}^{-1}}$, for example, represents the state $(\mathbf{x}, \mathbf{y}+1, \mathbf{z}, \mathbf{u}-1, \mathbf{v})$. We proceed by considering what is possible in a small period of time δt, considering separately the cases of no transitions, one exponential transition and one deterministic transition:

$$
\begin{aligned}
&\mathbb{P}\{\boldsymbol{S}_{t+\delta t} = \boldsymbol{s}\}\\
&\quad = \mathbb{P}(\{\boldsymbol{S}_t = \boldsymbol{s}\} \cap \{0 \text{ trans. in } [t, t+\delta t]\}) && (1)\\
&\quad + \mathbb{P}(\{\boldsymbol{S}_{t+\delta t} = \boldsymbol{s}\} \cap \{1 \text{ exp. in } [t, t+\delta t]\}) && (2)\\
&\quad + \mathbb{P}(\{\boldsymbol{S}_t = \boldsymbol{s}_{y^{+1},z^{-1}}\} \cap \{1 \text{ det. in } [t, t+\delta t]\}) && (3)\\
&\quad + o(\delta t)
\end{aligned}
$$

Eq. (2) evaluates in the usual manner as:

$$
\begin{aligned}
&\delta t(\mathbb{P}\{\boldsymbol{S}_t = \boldsymbol{s}_{\mathbf{x}^{+1},\mathbf{y}^{-1}}\}\lambda(\mathbf{x}+1) + \mathbb{P}\{\boldsymbol{S}_t = \boldsymbol{s}_{\mathbf{y}^{+1},\mathbf{x}^{-1}}\}\rho(\mathbf{y}+1)\\
&\quad + \mathbb{P}\{\boldsymbol{S}_t = \boldsymbol{s}_{\mathbf{z}^{+1},\mathbf{x}^{-1}}\}\rho(\mathbf{z}+1) + \mathbb{P}\{\boldsymbol{S}_t = \boldsymbol{s}_{\mathbf{v}^{+1},\mathbf{u}^{-1}}\}\lambda(\mathbf{v}+1)\\
&\quad + \mathbb{P}\{\boldsymbol{S}_t = \boldsymbol{s}_{\mathbf{y}^{+1},\mathbf{u}^{-1}}\}\frac{\beta(\mathbf{y}+1)(\mathbf{u}-1)}{N}\\
&\quad + \mathbb{P}\{\boldsymbol{S}_t = \boldsymbol{s}_{\mathbf{u}^{+1},\mathbf{v}^{-1}}\}\rho(\mathbf{u}+1)) + o(\delta t) && (4)
\end{aligned}
$$

By the Markov property, Eq. (1) can be written as:

$$
\begin{aligned}
&\mathbb{P}\{\boldsymbol{S}_t = \boldsymbol{s}\}\mathbb{P}\{0 \text{ exp. in } [t, t+\delta t] \mid \boldsymbol{S}_t = \boldsymbol{s}\}\\
&\qquad \times \mathbb{P}\{0 \text{ det. in } [t, t+\delta t] \mid \boldsymbol{S}_t = \boldsymbol{s}\}
\end{aligned}
$$

which is just:

$$
\begin{aligned}
&\mathbb{P}(\{\boldsymbol{S}_t = \boldsymbol{s}\} \cap \{0 \text{ det. in } [t, t+\delta t]\})\\
&\quad - \mathbb{P}\{\boldsymbol{S}_t = \boldsymbol{s}\}\delta t(\lambda\mathbf{x} + \rho\mathbf{y} + \rho\mathbf{z} + \frac{\beta\mathbf{y}\mathbf{u}}{N} + \lambda\mathbf{v} + \rho\mathbf{u})\\
&\qquad \times \mathbb{P}\{0 \text{ det. in } [t, t+\delta t] \mid \boldsymbol{S}_t = \boldsymbol{s}\} + o(\delta t) && (5)
\end{aligned}
$$

Multiplying by $\mathbf{y}$ and then summing each of Eqs. (3), (4) and (5) over all states $\boldsymbol{s}$, we may obtain:

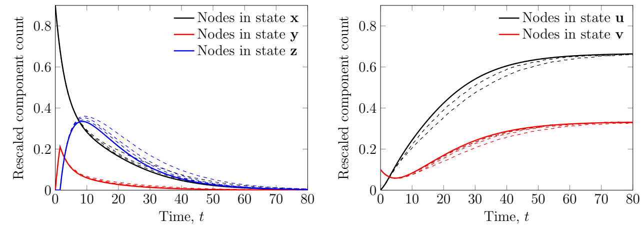
$$
\begin{aligned}
&\lim_{\delta t \to 0} \frac{\mathbb{E}[\mathbf{y}_{t+\delta t}] - \mathbb{E}[\mathbf{y}_t]}{\delta t} =\\
&\quad \lambda\mathbb{E}[(\mathbf{y}_t+1)\mathbf{x}_t] + \rho\mathbb{E}[(\mathbf{y}_t-1)\mathbf{y}_t] + \rho\mathbb{E}[\mathbf{y}_t\mathbf{z}_t]\\
&\quad + \frac{\beta}{N}\mathbb{E}[(\mathbf{y}_t-1)\mathbf{y}_t\mathbf{u}_t] + \lambda\mathbb{E}[\mathbf{y}_t\mathbf{v}_t] + \rho\mathbb{E}[\mathbf{y}_t\mathbf{u}_t]\\
&\qquad - \lim_{\delta t \to 0} \mathbb{P}(\{1 \text{ det. in } [t, t+\delta t]\})/\delta t\\
&\qquad - \lambda\mathbb{E}[\mathbf{y}_t\mathbf{x}_t] - \rho\mathbb{E}[\mathbf{y}_t^2] - \rho\mathbb{E}[\mathbf{y}_t\mathbf{z}_t]\\
&\qquad - \frac{\beta}{N}\mathbb{E}[\mathbf{y}_t^2\mathbf{u}_t] - \lambda\mathbb{E}[\mathbf{y}_t\mathbf{v}_t] - \rho\mathbb{E}[\mathbf{y}_t\mathbf{u}_t] && (6)
\end{aligned}
$$

Note that:

$$
\begin{aligned}
&\lim_{\delta t \to 0} \mathbb{P}(\{1 \text{ det. in } [t, t+\delta t]\})/\delta t = \mathbf{1}_{t \geq \gamma} \lim_{\delta t \to 0}\\
&\quad \frac{1}{\delta t}\mathbb{P}(\{x \to y \text{ in } [t-\gamma, t-\gamma+\delta t],\ \not\to u \text{ and } \not\to x \text{ before } t\})
\end{aligned}
$$

Then by conditioning on a transition $x \to y$ in $[t-\gamma, t-\gamma+\delta t]$, we see that this is:

$$
\mathbf{1}_{t \geq \gamma}\lambda \exp(-\rho\gamma)\mathbb{E}[\mathbf{x}_{t-\gamma}]\mathbb{E}\left[\exp\left(-\int_{t-\gamma}^{t} \frac{\beta\mathbf{u}_s}{N}\,\mathrm{d}s\right)\right] \qquad (7)
$$



(a) Old nodes *(b)* Updated nodes

Figure 1: DDE approximation (solid line) compared with actual means for $N = 20$, 50 and 100 (dashed lines). Initial component proportions are $(0.9, 0, 0, 0, 0.1)$ and parameters are $\lambda = 0.2$, $\beta = 2.0$, $\rho = 0.1$ and $\gamma = 1.5$.

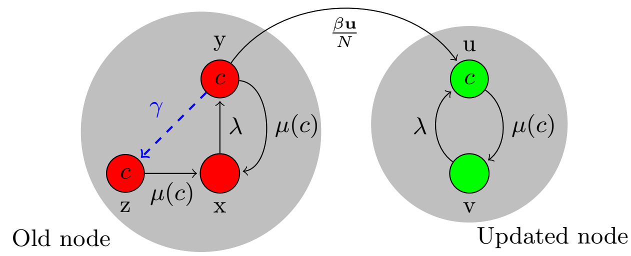


Figure 2: GSMP representation of the behaviour of a single node in the software update process model with generally-distributed on-times.

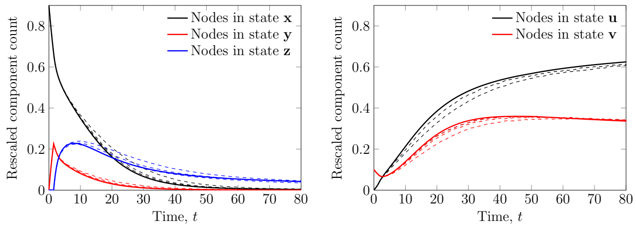
Finally, applying the approximation $\mathbb{E}[f(X)] \approx f(\mathbb{E}[X])$ to Eqs. (6) and (7) for random variables X which is common to the derivation of mean-field equations, we may derive the following DDE whose solution $\tilde{\mathbf{y}}(t)$ approximates $\mathbb{E}[\mathbf{y}_t]$:

$$
\begin{aligned}
\dot{\tilde{\mathbf{y}}}(t) = &-\rho\tilde{\mathbf{y}}(t) - \frac{\beta}{N}\tilde{\mathbf{y}}(t)\tilde{\mathbf{u}}(t) + \lambda\tilde{\mathbf{x}}(t)\\
&- \mathbf{1}_{t \geq \gamma}\lambda \exp(-\rho\gamma)\tilde{\mathbf{x}}(t-\gamma)\exp\left(-\int_{t-\gamma}^{t} \frac{\beta\tilde{\mathbf{u}}(s)}{N}\,\mathrm{d}s\right)
\end{aligned}
$$

DDEs for the other four quantities can be derived similarly. These can then be integrated numerically using, for example, the `dde23` routine in `MATLAB`®. Figure 1 compares the DDE approximation with the actual values computed by stochastic simulation. All component counts are rescaled by the component population size N and we see that under this regime, we appear to have mean-field convergence.

3. GENERAL DISTRIBUTIONS

In this section we show briefly how the approach of the last section can be extended to consider generally-timed transitions. In particular, we consider the same model as in the previous section but where the time for which a node (either old or updated) remains on is given by an arbitrary distribution with, say, cumulative distribution function $F(t)$, probability density function $f(t) := \frac{d}{dt}F(t)$ and instantaneous rate function $\mu(t) := \frac{f(t)}{1-F(t)}$. A representation of the behaviour of a single node as a GSMP is given in Figure 2. The deterministic timeout behaviour is also kept in the model and represented by the dashed transition. In all of the states y, z and u, a continuous state variable c is required to keep track of the elapsed time since the node originally came on. The value of c is maintained over transitions between these three states so that if a node is updated or the deterministic transition fires, this clock is *not* reset.



(a) Old nodes *(b)* Updated nodes

Figure 3: DDE approximation (solid line) compared with actual means for $N = 20$, 50 and 100 (dashed lines). Initial component proportions are $(0.9, 0, 0, 0, 0.1)$ and parameters are $\lambda = 0.2$, $\beta = 2.0$ and $\gamma = 1.5$. The on-time has a Pareto distribution with scale parameter $t_m = 1.5$ and shape parameter $\alpha = 0.9$, so that $F(t) = 1 - (\frac{t_m}{t})^\alpha$ for $t \geq t_m$ and $F(t) = 0$ otherwise.

The mean-field equations may be derived by following a similar procedure to that of the previous section. The presence of general distributions introduces convolution integral terms which depend on the entire history of the solution, weighted appropriately by the corresponding density $f(t)$. For the sake of brevity, we will not give a detailed derivation of the mean-field equations here as we did in the previous section. Instead, we present the equation for $\tilde{\mathbf{y}}(t)$ (the largest equation in the system of five equations) and give heuristic justification for the terms which constitute its right-hand side:

$$\dot{\tilde{\mathbf{y}}}(t) = \overbrace{-\frac{\beta}{N}\tilde{\mathbf{y}}(t)\tilde{\mathbf{u}}(t) + \lambda\tilde{\mathbf{x}}(t)}^{\text{drift due to exponential transitions}}$$
$$\overbrace{-\int_0^{\gamma\wedge t} \underbrace{\lambda\tilde{\mathbf{x}}(t-s)}_{x\to y \text{ rate at } t-s} \overbrace{\exp\left(-\int_{t-s}^{t} \frac{\beta\tilde{\mathbf{u}}(b)}{N}\,\mathrm{d}b\right)}^{\text{prob. } y\not\to u \text{ in } [t-s,t]} f(s)\,\mathrm{d}s}^{\text{drift due to transitions } y\to x}$$
$$\overbrace{-\mathbf{1}_{t\geq\gamma} \underbrace{\lambda\tilde{\mathbf{x}}(t-\gamma)}_{x\to y \text{ rate at } t-\gamma} \overbrace{(1-F(\gamma))}^{\text{prob. } y\not\to x \text{ in } [t-\gamma,t]} \underbrace{\exp\left(-\int_{t-\gamma}^{t} \frac{\beta\tilde{\mathbf{u}}(b)}{N}\,\mathrm{d}b\right)}_{\text{prob. } y\not\to u \text{ in } [t-\gamma,t]}}^{\text{drift due to transitions } y\to z}$$

This system can be solved numerically by adapting standard routines such as `dde23` in `MATLAB`® to keep track of the entire history of the solution (suitably discretised). Figure 3 compares the DDE approximation with the actual values computed by stochastic simulation. Here we have used a Pareto-distributed on-time. As before, all component counts are rescaled by the component population size N and we see again that we appear to have mean-field convergence.

More specifically, we have constructed in this section essentially a system of non-linear *Volterra integro-differential equations* of *convolution type* [1]. The naïve approach to computing solutions to such equations numerically requires the entire history of the discretised solution to be kept in memory and the recomputation of the convolution integral terms at each time step. This results in the computation time being dependent quadratically on the number of discretisation steps, say N_t. In such cases, it is not entirely clear whether or not this approach is more or less efficient than first fitting phase-type approximations to general holding times in the original model and then solving instead the resulting much larger system of standard mean-field ODEs.

However, recent work of Lubich [9] proposes a fast algorithm to compute convolution integrals $\int_0^t f(t-s)g(s)\mathrm{d}s$ where the function $g(t)$ is not known in advance but computed from the value of the convolution at time t. It has $O(N_t \log(N_t))$ time complexity and can be used to construct Runge–Kutta [5] methods capable of solving equations such as those constructed in this section numerically also in $O(N_t \log(N_t))$ time. Investigating how these approaches compare with the complexity of a sufficiently accurate phase-type approximation followed by traditional Markovian mean-field analysis is the subject of our current research.

4. CONCLUSION

In this paper we have shown that delay differential equations (DDEs) provide a promising avenue for the efficient analysis of massively-parallel systems which include generally-timed transitions. In addition to that already discussed, future work includes extending these techniques to obtain higher-order moments (cf. [7]) and rewards (cf. [12]) from such models. Additionally, formal proof of mean-field convergence and error bounds would provide greater confidence in the approach.

5. REFERENCES

[1] C. Baker. A perspective on the numerical treatment of Volterra equations. *Journal of Computational and Applied Mathematics*, 125(1-2):217–249, Dec. 2000.

[2] R. Bakhshi, L. Cloth, W. Fokkink, and B. R. Haverkort. Mean-field framework for performance evaluation of push-pull gossip protocols. *Perf. Evaluation*, 68(2):157–179, Feb. 2011.

[3] M. Benaïm and J.-Y. Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65(11-12):823–838, Nov. 2008.

[4] A. Bobbio, M. Gribaudo, and M. Telek. Analysis of Large Scale Interacting Systems by Mean Field Method. In *Fifth International Conference on Quantitative Evaluation of Systems*, pages 215–224. IEEE, Sept. 2008.

[5] G. Capobianco, D. Conte, I. Del Prete, and E. Russo. Fast Runge–Kutta methods for nonlinear convolution systems of Volterra integral equations. *BIT Numerical Mathematics*, 47(2):259–275, Feb. 2007.

[6] D. R. Cox. The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(03):433–441, July 1955.

[7] R. A. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian process algebra. *Theoretical Computer Science*, 411(22-24):2260–2297, May 2010.

[8] C. Lindemann and A. Thümmler. Transient analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions. *Performance Evaluation*, 36-37(1):35–54, Aug. 1999.

[9] C. Lubich and A. Schädle. Fast Convolution for Nonreflecting Boundary Conditions. *SIAM Journal on Scientific Computing*, 24(1):161, Jan. 2002.

[10] K. Matthes. Zur Theorie der Bedienungsprozesse. In *Transactions of the 3rd Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, pages 512–528, 1962.

[11] A. Puliafito, M. Scarpa, and K. S. Trivedi. Petri nets with k simultaneously enabled generally distributed timed transitions. *Performance Evaluation*, 32(1):1–34, Feb. 1998.

[12] A. Stefanek, R. A. Hayden, and J. T. Bradley. Fluid analysis of energy consumption using rewards in massively parallel Markov models. In *2nd ACM/SPEC International Conference on Perf. Engineering (ICPE)*, pages 121–132, Karlsruhe, 2011.