Stochasticity of probabilistic systems: Analysis methodologies case-study*

Anwitaman Datta, Martin Hasler, Karl Aberer
{anwitaman.datta, martin.hasler, karl.aberer}@epfl.ch
Ecole Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
CH-1015 Lausanne, Switzerland

Abstract

We do a case study of two different analysis techniques for studying the stochastic behavior of a randomized system/algorithms: (i) The first approach can be broadly termed as a mean value analysis (MVA), where the evolution of the mean state is studied assuming that the system always actually resides in the mean state. (ii) The second approach looks at the probability distribution function of the system states at any time instance, thus studying the evolution of the (probability mass) distribution function (EoDF).

1 Introduction

Designing large-scale distributed systems in a decentralized setting often relies heavily on randomized algorithms and self-organization. There are several interesting and critical properties of such large-scale probabilistic systems that need to be understood in order to design and assess them. Some important and interesting properties include: (i) The expected behavior of the system over time. (ii) The (degree of) deviation of the system from its expected behavior. (iii) Whether the system converges to some steady state. There are three important dimensions that together account for properly describing the system - (i) *time*, (ii) *design parameters* and (iii) *system state* (specific properties being observed in order to assess the system).

The system properties can generally be studied by modeling the system as a stochastic process (often Markovian). We observe that two different analysis techniques can be found in the literature for studying the stochastic behavior of a randomized system/algorithms: (i) The first approach can be broadly termed as a *mean value analysis (MVA)*, where the evolution of the mean state is studied assuming that the system always actually resides in the mean state. (ii) The second approach looks at the probability distribution function of the system states at any time instance, thus studying the *evolution of the (probability mass) distribution function (EoDF)*.

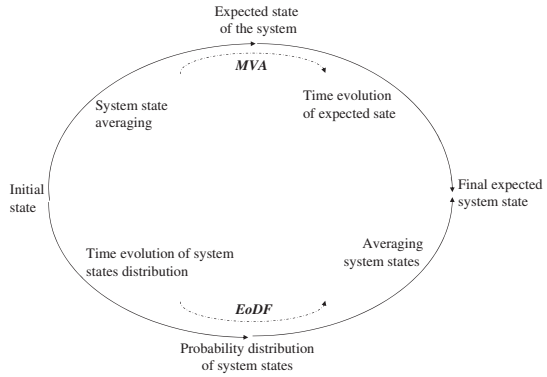
The former approach (MVA) simplifies the analysis and is a convenient tool for system designers to understand the expected behavior of the system, but at the cost of losing potentially important information like the deviation, the stability and the convergence of the system. This is because MVA collapses the problem into two dimensions - time and design parameters, with the system state assumed to have a singular value (corresponding to the mean) at any point of time. That is to say, the system state is averaged first, and then the time evolution of such an average-state approximation of the system is studied. Because of the reduction of the whole system states into a single (averaged) value, the resulting analysis is simpler, and for a desired resulting system state, the design parameters can be determined analytically.

EoDF on the other hand results in a more complicated and larger set of equations since it looks into the time evolution of the probability distribution of the system states for any choice of parameters. Thus solving these equations is computationally more intensive. The reward is that it yields detailed information of the deviation of the system behavior from the mean behavior.

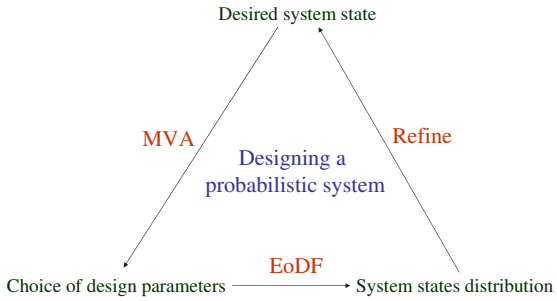
If the time evolution and averaging commute (Figure 1(a)), then the expectation values obtained from the EoDF and MVA analyses concur. In the example studied in this paper, it is the case up-to a certain time. If the commutation does not hold, then MVA does not give the correct mean behavior. It may however still be a good approximation. Only EoDF analysis can validate this, and thus MVA and EoDF can still both be used together (Figure 1(b)) to efficiently design, validate (and iteratively refine) a proba-

*The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 and supported by the Swiss National Funding Agency OFES as part of the European project Evergrow No 001935.

1-4244-0030-9/05/$20.00 ©2005 IEEE



(a) MVA vs. EoDF



(b) Iterative re nement for system designing (reducing computation complexity)

Figure 1. Probabilistic systems analysis methodologies: MVA and EoDF

bilistic system.

This paper is a case-study of the use of these analysis techniques in the study of an algorithm used in load-balanced structured overlay network construction for data-oriented applications[1].

2 Case-study: Proportional partitioning with referential integrity

In recent years self-organization has been both observed in and used to design decentralized peer-to-peer networks in diverse contexts: including *evolution* of the networks (power-law networks [3], structured overlays [1], forming groups [6]), *search* (in power-law networks [2], in semi-structured networks [4]) to name a few.

Here, we are speci cally interested in what are called structured overlay networks. Simply put, the basic idea of structured overlays is to (i) partition a (key-)space among peers, so that peers are responsible for some speci c partitions, and to (ii) embed a graph/routing network among these peers (respective partitions), so that any partition can be reached starting from any peer ef ciently and reliably. One possible way to build an overlay network rapidly is to recursively repartition the key-space among the peers. In that context, in order to achieve load-balancing at individual peers in presence of arbitrary load-distribution over the key-space, this partitioning process also needs to take into account the load-skew. In the following, we focus on the single bisection step of a skewed key-space. This process can then be repeated recursively in order to construct an overlay network [1]. Next we describe the partitioning problem:

Consider a set P of $n+1$ peers which hold data keys from key space K. The space K is partitioned into two parts, lets say 0 and 1, such that the load measured in number of data keys related to the partitions, l_0 and l_1 are p and $1-p$. We assume w.l.o.g. in the following that $0 \leq p \leq \frac{1}{2}$. Then the partitioning that we would ideally like to achieve should have the following properties:

Proportional replication: Each peer has to decide for one of the two partitions such that (in expectation) a fraction p of the peers decide for 0 and a fraction $1-p$ for 1. Thus the workload becomes uniformly distributed among the peers, meeting the load-balancing criteria in the resulting overlay.

Referential integrity: Each of the peers has to encounter during the process at least one peer that decided for the other partition. Thus the peers have the necessary information to construct a routing structure for delegating requests for keys they are no longer associated with, and hence facilitating overlay routing.

In the rest of this paper we focus solely on the analysis of a distributed mechanism to address the problem of proportional partitioning with referential integrity. This single partitioning process is in turn recursively used in the sub-partitions so formed in order to construct a structured overlay [1] - the details of which is out of the scope of this paper.

Adaptive eager partitioning algorithm

The decentralized algorithm we design (AEP: Adaptive eager partitioning) to achieve the partitioning constraints assumes that each peer can initiate interaction with any ran-

[1] In this paper we only focus on the analysis of the building block of our overlay network construction algorithms. Refer to [1] for more details on the overlay network (construction) itself.

dom peer. The speci cs leading to the design of the algorithm and its use in overlay network construction are to be found in [1], so we don't go into the details and instead keep the exposition focussed on the analysis techniques. The AEP algorithm is as follows:

1. Each undecided peer initiates interactions with a uniformly randomly selected peer until a decision is reached. Selecting peers uniformly at random is a non-trivial problem in itself which we solve by a variant of random walks.

2. If the contacted peer is undecided the peers perform a balanced split with probability $0 \leq \alpha(p) \leq 1$ and maintain references to each other.

3. If the contacted peer has already decided for 0 then the contacting peer decides for 1 and maintains a reference to the contacted peer.

4. If the contacted peer has already decided for 1 then the contacting peer decides for 0 with probability $0 \leq \beta(p) \leq 1$ and with probability $1 - \beta(p)$ for 1. In the rst case it maintains a reference to the contacted peer. In the second case it obtains a reference to a peer from the other partition from the contacted peer.

While the referential integrity constraint is easily met, in order to meet the proportional partitioning constraint, one needs to choose parameters $\alpha(p)$ and $\beta(p)$ appropriately.

3 Analysis

We model the interactions as a Markovian process in discrete time t. Lets say there are $n+1$ peers that perform the partitioning. Furthermore, consider n_t^0 and n_t^1 being the number of peers that have already decided for the partitions 0 and 1 respectively, while the remaining peers are undecided[2]. The states of the Markov chain are characterized by these two numbers since we don't have to distinguish among the individual peers. Formally:

$S = \{\begin{pmatrix} n^0 \\ n^1 \end{pmatrix} | n^0, n^1 \geq 0; n^0 + n^1 \leq n+1\}$.

All states where all peers have decided are absorbing, and all other states are transient [5].

3.1 Mean value analysis [MVA]

Initially all peers are undecided, i.e., $n_0^0 + n_0^1 = 0$. At the end of the process, for some time step t_f the partitioning process is expected to nish, s.t. $n_{t_f}^0 + n_{t_f}^1 = n+1$. Along with these boundary conditions, we have the time evolution of the mean values given as follows:

$$\overline{n}_{t+1}^0 = \overline{n}_t^0 + \frac{1}{n}(n - \overline{n}_t^0 - \overline{n}_t^1)\alpha + \frac{\overline{n}_t^1}{n}\beta \quad (1)$$

$$\overline{n}_{t+1}^1 = \overline{n}_t^1 + \frac{1}{n}(n - \overline{n}_t^0 - \overline{n}_t^1)\alpha + \frac{\overline{n}_t^0}{n} + \frac{\overline{n}_t^1}{n}(1-\beta) \quad (2)$$

Note that these equations are linear and therefore they can be solved explicitly. In particular, if we sum the two equations, we get the mean value $\overline{m}_{t+1} = \overline{m}_t(1 - \frac{2\alpha-1}{n}) + 2\alpha$. Solution for this recursion for the initial condition of $\overline{m}_0 = 0$ is:

$\overline{m}_t = n\frac{2\alpha}{2\alpha-1}(1 - (1 - \frac{2\alpha-1}{n})^t)$.

From this expression we can calculate the termination time t_f.

$$t_f = n\frac{ln(2\alpha)}{2\alpha - 1} \quad (3)$$

If we evaluate the solution of the linear equations 1&2 at this value of t_f, we obtain, in the spirit of MVA, the final values of $\overline{n}^0$ and $\overline{n}^1$, and therefore p as an explicit function of α and β. For design purposes this function can be numerically inverted in order to obtain either of α or β as a function of the other and the design parameter p.

3.2 Evolution of (probability) distribution function [EoDF]

The time evolution of the probability mass function - the distribution of n^0 and n^1 can be given as:

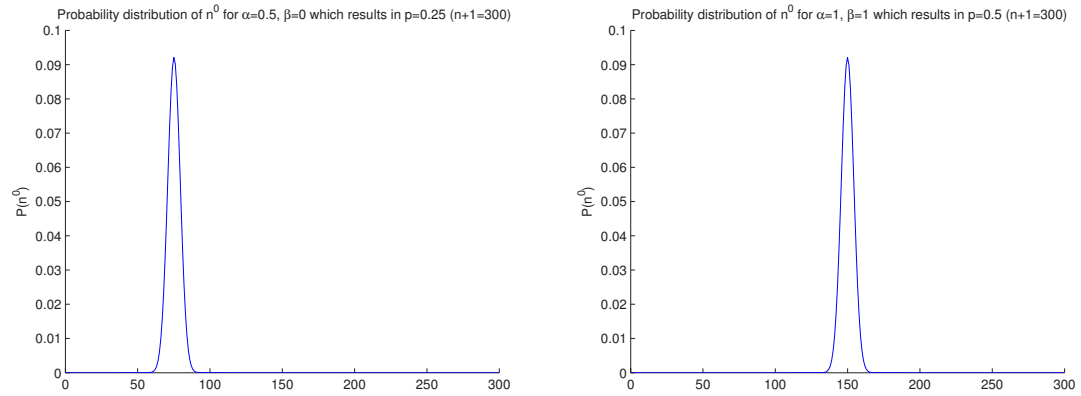
$$\begin{aligned} P\left[\begin{pmatrix} n^0 \\ n^1 \end{pmatrix}, t+1\right] &= P\left[\begin{pmatrix} n^0 - 1 \\ n^1 \end{pmatrix}, t\right]\frac{n^1}{n}\beta \\ &+ P\left[\begin{pmatrix} n^0 \\ n^1 - 1 \end{pmatrix}, t\right]\frac{n^0 + (n^1 - 1)(1-\beta)}{n} \\ &+ P\left[\begin{pmatrix} n^0 - 1 \\ n^1 - 1 \end{pmatrix}, t\right]\frac{1}{n}\alpha \\ &+ P\left[\begin{pmatrix} n^0 \\ n^1 \end{pmatrix}, t\right]\frac{1}{n}(1-\alpha) \end{aligned} \quad (4)$$

for $n^0 + n^1 \leq n$ with the last term replaced by $P\left[\begin{pmatrix} n^0 \\ n^1 \end{pmatrix}, t\right]$ for $n^0 + n^1 = n+1$.

In the expression above, we assume that $P\left[\begin{pmatrix} x \\ y \end{pmatrix}\right] = 0$ if either $x < 0$ or $y < 0$.

This equation, along with the initial condition $P\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, 0\right] = 1$ and $P\left[\begin{pmatrix} n^0 \\ n^1 \end{pmatrix}, 0\right] = 0$ for all other cases, allows to determine the probability distribution function at any time t for any choice of n, α, β. We are of-course primarily interested in the probability mass function of the states at the end of the partitioning process. It turns out that for $n \to \infty$ the expectation value for n^0 and n^1 follow the equations (1)&(2) until t_f and remain

[2]We do not use t where the context is unambiguous.



(a) (b)

Figure 2. Probability distribution function obtained using EoDF for α and β chosen for desired $p = \frac{n^0}{n+1}$ values.

constant thereafter. Thus to say the expectation values of the EoDF and the MVA analyses match in our case study.

Note that if the system were to lead to non-linear equations (as may be the case for some some other systems) then the MVA analysis might have led to a different result than the EoDF analysis. However solving the MVA equations numerically is computationally much cheaper than solving the equations obtained from EoDF analysis. This is because for the MVA analysis, there is only one variable per system state, representing its mean value. In contrast, for EoDF we need to deal with the whole distribution of each of the states, which makes solving the EoDF based equations computationally expensive. However EoDF allows in addition to obtain the higher order moments of the system states n^0 and n^1 as a function of the design parameters α and β, apart from the probability distribution function itself, which is calculated from equation 4. Figure 2 shows the probability distribution function obtained for various desired values of p for a total peer population $n + 1 = 300$ where the design parameters α and β are chosen by solving the MVA based equations.

4 Conclusions

Large scale systems are often modeled as probabilistic systems. Both the analysis methodologies discussed in this paper are often used, but mostly without discerning the subtlety of MVA, which is only an approximation of the actual system behavior. The appropriate way to study such systems is by following the time evolution of the distribution function of the states. While such an exhaustive study is precise and provides better insight into the systems properties (like higher order moments) it is also relatively complex. On the other hand, approximating the system by following how a singular state (representative of the mean) would evolve over time is relatively simpler and convenient for a system designer to make approximate design choices.

In the example discussed in this paper, it turns out that a design based on the expectation values of the EoDF analysis is identical to the MVA (because of linear intra/inter-dependance of system states and parameters). In general, however, the time evolution of the actual expectation values will be more complex (taking the expectation from the probability distribution obtained with EoDF) than the MVA analysis, and the results will not necessarily be identical. In such a case, because of its relative simplicity MVA could still be used to get a first approximate idea of the influence of the system parameters on system behavior and subsequent EoDF analysis for the already chosen design parameters would allow to assess the accuracy of MVA, to correct it (fine-tune the design parameters) if necessary, and to estimate the deviation from the mean values/expectations. This design cycle is briefly illustrated in Figure 1(b).

References

[1] K. Aberer, A. Datta, M. Hauswirth, and R. Schmidt. Indexing data-oriented overlay networks. In *31st International Conference on Very Large Data Bases (VLDB)*, 2005.

[2] L.A. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman. Search in power-law networks. *Physical Review E*, 64:046135, 2001.

[3] A-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[4] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2001.

[5] W. Feller. *An Introduction to Probability Theory and Its Applications.* Wiley series in probability and mathematical statistics, 1968.

[6] J. Ledlie, J. Taylor, L. Serban, and M. Seltzer. Self-organization in peer-to-peer systems. In *10th EW SIGOPS*, 2002.