A Practical Methodology for Defining Histograms for Predictions and Scheduling

Jennifer M. Schopf*
Computer Science Department
Northwestern University
jms@cs.nwu.edu

February 28, 1999

Abstract

Current distributed parallel platforms can provide the resources required to execute a scientific application efficiently. However, when these platforms are shared by multiple users, the performance of the applications using the system may be impacted in dynamic and often unpredictable ways. Performance prediction becomes increasingly difficult due to this dynamic behavior. Even performance modeling techniques that are built specifically for distributed parallel systems often require parameterization by single (point) values. However, in shared environments, point values may provide an inaccurate representation of application behavior due to variations in resource performance.

This paper address the use of **practical histogram stochastic values** *to parameterize performance models. Whereas a point value provides a single value representation of a quantity, a stochastic value provides a set of possible values to represent a range of likely behavior. In previous work we investigated using either an upper bound and lower bound, or a normal distribution to represent stochastic data. In this work we examine using a combination of those methods, namely histograms, to represent this data. We define a practical definition in order to use histograms in a production setting, and then show how predictions of this nature are more or less suited to application performance.*

1 Motivation

Current distributed parallel platforms can provide the resources required to execute a scientific application efficiently. However, when these platforms are shared by multiple users, the performance of the applications using the system may be impacted in dynamic and often unpredictable ways. In order to obtain good performance, accurate performance prediction models for distributed parallel systems are needed. Most

*Partially supported by NASA GSRP Grant #NGT-1-52133 and Darpa grant #N660001-97-C-8531.

performance prediction models use parameters to describe system and application characteristics such as bandwidth, available CPU, message size, operation counts, etc. Model parameters are generally represented as a single likely value, which we refer to as a **point value**. For example, a point value for bandwidth might be 7 Mbits/second.

In practice, point values are often a best guess, an estimate under ideal circumstances, or a value that is accurate only for a given time-frame. In some situations it may be more accurate to represent system and application characteristics as a *range* of possible values; for example, bandwidth might be reported as varying between 6 and 8 Mbits/second. We refer to such values as **stochastic values**. Whereas a point value gives a single value for a quantity, a stochastic value gives a set of values, possibly weighted by probabilities, to represent a range of likely behavior [TK93].

One way to represent stochastic values is by using histograms, which consist of a set of intervals with associated probabilities. In this paper we define histograms and discuss alternative ways to construct them, including out own practical definition for use in performance prediction techniques. In the full paper we will also give a brief over view of a modeling technique called structural modeling that can be parameterized by stochastic information, give further examples, and discuss the arithmetic needed to combine histograms for use in practical models.

2 Formal Definitions for Histograms

A stochastic value X may be specified using a one-dimensional histogram $H(X)$ as follows:

$$\begin{aligned} H(X) = \quad & X_1, X_2, ...X_m \\ X_1 \quad & = [\underline{x_1}, \overline{x_1}] : p_1 \\ X_2 \quad & = [\underline{x_2}, \overline{x_2}] : p_2 \\ & \vdots \\ X_m \quad & = [\underline{x_m}, \overline{x_m}] : p_m \end{aligned} \tag{1}$$

where $\sum_{i=1}^{m} p_i = 1$. Each X_i entry in the definition of $H(X)$ is a pair consisting of an interval $[\underline{x_i}, \overline{x_i}]$, and an associated probability, p_i. Graphically, this is depicted in Figure 1. The histogram approach provides more information about a set of values than a single interval, but less information than a distribution since the distribution of probabilities over each sub-interval in the histogram is assumed to be uniform.

One of the primary difficulties in defining histograms for stochastic values is the determination of the appropriate intervals and probabilities for each histogram. There are a number of guidelines for defining histograms in the literature, but in most cases these concentrate on ways to *display* the information, as opposed to *using* the information as a summary technique in a practical setting.

In the most closely related work to our research [LH95], histograms are defined initially based on the number of intervals provided by the user. The intervals are then split, using interval splitting [MR95, Ske74] and a user supplied procedure to produce more subintervals for intervals with high probabilities and fewer for intervals of the same width but lower probability. This procedure is iterated until the changes that result

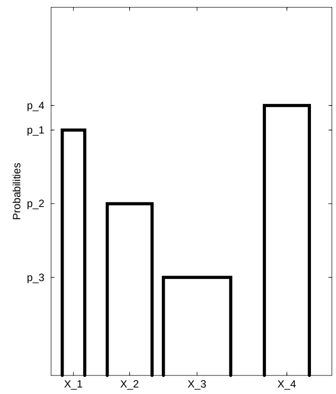

Figure 1. Example histogram.

by splitting the intervals no longer affects the resulting prediction. For their purposes, "Computational complexity is reasonable given that the number of parameters specified as histograms is not too high" [LMH96], however, this is not practical for our setting.

3 Practical Histograms

When defining histograms for use in predictions for run-time scheduling, we needed to define an approach that would use few user supplied parameters, would be computationally efficient, and yet would adequately capture the behavior of the system being described. For example, one model parameter needed in most contentious environment is a measurement of the available CPU. In our setting, we had available to us a time series of previous available CPU measurements supplied by an online tool called the Network Weather Service [Wol96, Wol97].

We used a two-pass technique and limited the number of individual intervals within the histogram to 5. Experimentally, this value worked well. Future work involves dynamically selecting the number of interval bins to use. The first pass over the data was used to evaluate an upper bound and a lower bound, and to determine the size of the individual interval bins. The second pass was used to sort the data into the interval bins, and define the histogram for the stochastic data.

4 Predictions Using Histograms

To use stochastic information in the form of histograms to predict application behavior, first the parameters must be processed and put into a format that a performance model can understand. That performance model must be solved, and supplies a scheduling technique with prediction information in order to make good scheduling decisions. The resulting prediction will also be a histogram. This is depicted in Figure 2.

Figure 3 is a sample prediction for an SOR code over four Sparc workstations under variable load from other users. The full paper will have sets of these for 3-4 applications under differing workloads.

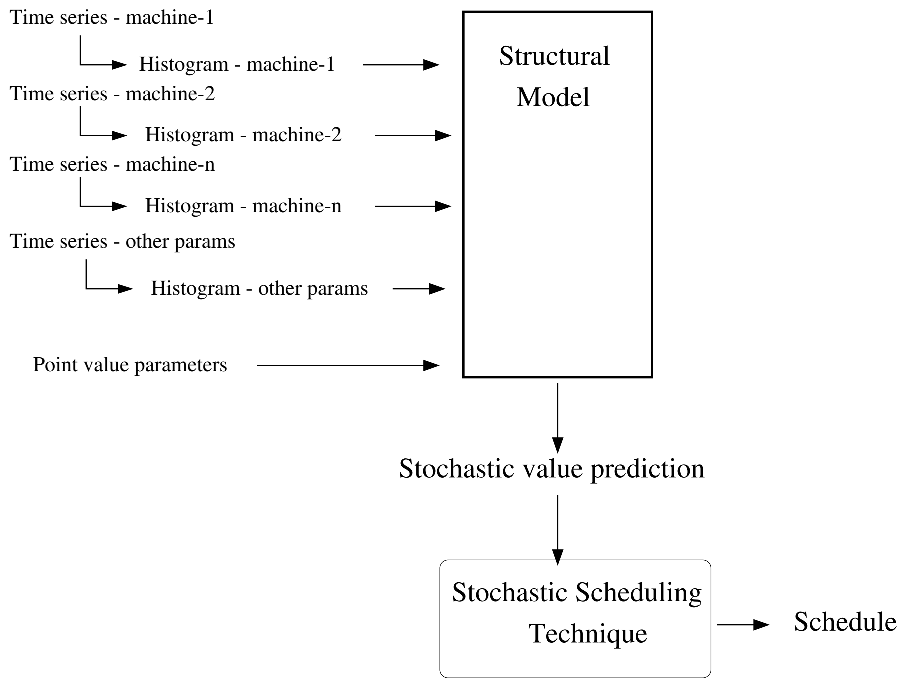

Figure 2. Steps to make a stochastic prediction.

5 Discussion

When combining stochastic values represented as histograms, it is assumed that each partial result is uniformly distributed. If this assumption does not hold, the intervals can be subdivided using interval splitting [MR95, Ske74], creating smaller intervals that are more likely to fit this assumption. Even if the intervals are relatively small, uniformity may not accurately capture the distribution of the target data.

In addition, the histogram representation assumes that each stochastic value is uncorrelated in order to determine the probabilistic percentages associated with each partial result. If the stochastic values are correlated, weights reflecting the correlation must be supplied, a difficult task in practice.

In previous work [Sch98, SB99, SB97] we investigated representing stochastic values as distributions and intervals. Each approach has advantages and disadvantages. Intervals are the most easily defined since the minimum and maximum value in a data set for a stochastic value are easy to determine. However, outliers in the data can affect the size of the interval, and no details about the shape of the data are included in a simple range. Histograms allow the shape of a stochastic value to be elucidated, and lend themselves to the grouping of values, but are difficult to define in a practical setting. Distributions can be defined using well understood metrics for the data, but in order to be tractable arithmetically, we must assume that the associated data fits a well-defined and computationally efficient family of distributions, such as the family of normal distributions, which is not always a valid assumption.

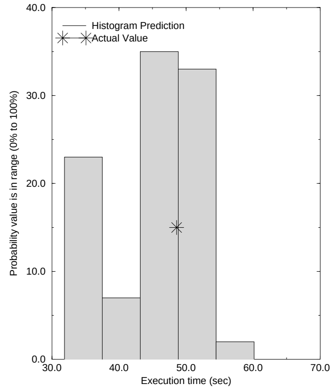

Figure 3. Sample Histogram Prediction.

References

[LH95] Johannes Lüthi and Günter Haring. Mean value analysis for queuing network models with intervals as input parameters. Technical Report TR-950101, Institute of applied science and information systems, university of Vienna, July 1995.

[LMH96] Johannes Lüthi, Shikharesh Majumdar, and Günter Haring. Mean value analysis for computer systems with variabilities in workload. In *Proceedings of the IEEE International Computer Performance and Dependability Symposium, IPDS '96*, 1996.

[MR95] Shikharash Majumdar and Ravathy Ramadoss. Interval-based performance analysis of computing systems. In *Proceedings of the Third International Workshop on Model Analysis and Simulation of Computer and Telecommunication Systems*, 1995.

[SB97] Jennifer M. Schopf and Francine Berman. Performance prediction in production environments. In *Proceedings of IPPS/SPDP '98*, 1997.

[SB99] Jennifer M. Schopf and Francine Berman. Using stochastic intervals to predict application behavior on contended resources. In *Submitted to the Workshop on Advances in Parallel Computing Models*, 1999. Also avilable as Northwestern Univerisy, Computer Science Department Technical Report CS-99-1, or `http://www.cs.nwu.edu/~jms/Pubs/TechReports/hilo.ps`.

[Sch98] Jennifer M. Schopf. *Performance Prediction and Scheduling for Parallel Applications on Multi-User Clusters*. PhD thesis, University of California, San Diego, 1998. Also available as UCSD CS Dept. Technical Report, Number CS98-607, `http://www.cs.nwu.edu/~jms/Thesis/thesis.html`.

[Ske74] Stig Skelboe. Computation of rational integer functions. *BIT*, 14, 1974.

[TK93] Howard M. Taylor and Samuel Karlin. *An Introduction to Stochastic Modeling*. Academic Press, 1993.

[Wol96] Rich Wolski. Dynamically forecasting network performance using the network weather service(to appear in the journal of cluster computing). Technical Report TR-CS96-494, UCSD, CSE Dept., 1996.

[Wol97] R. Wolski. Dynamically forecasting network performance to support dynamic scheduling using the network weather service. In *Proc. 6th IEEE Symp. on High Performance Distributed Computing*, August 1997.