Record Search Time Evaluation

K. Ko and T.G. Robertazzi
Department of Electrical and
Computer Eng.
SUNY at Stony Brook
Stony Brook, NY 11794
e-mail: tom@ece.sunysb.edu

Abstract — **Divisible scheduling theory [1] is used to solve for the expected time for searching for both single and multiple data records in parallel database architectures. The target architectures examined for illustrative purposes are the linear daisy chain and single level tree network with single and multiple installment load distribution.**

I. Introduction

Divisible loads are ones that can be arbitrarily partitioned among a number of processors. The use of divisible load modeling and associated analysis to evaluate parallel systems has a number of advantages. Linear and continuous modeling results in a tractable analysis. The underlying load allocation equations are deterministic so no probabilistic assumptions are made in this part of the analysis. Furthermore the model is generic enough that it can accommodate changes in technology and topology.

In this paper elegant expressions for the expected time to find both single and multiple records are found. The architectures investigated are a linear daisy chain of processor and a single level tree network. For the tree network both single installment and multi-installment load distribution is considered. The techniques described here can be used to model and solve for record search time on other architectures. This work is significant for demonstrating the power of divisible load scheduling theory for predicting search times.

II. Expected Time of Searching for a Single Record

A record position variable is described in terms of the uniform distribution and is denoted as $\mathbf{X}$. Since the dataset is normalized, a record is positioned between zero and one. Further, it is assumed that the dataset is very large. Thus, the distribution of a record position is regarded as continuous variable. Now, considering the quantity that must be described as the amount of time till a record is found on a given processor, this random variable denoted by $\mathbf{Y}$ depends on and can be obtained from the random variable of a record position, $\mathbf{X}$.

The closed form of the expected time of searching for a record is:

$$\mathrm{E}\left[\mathbf{Y}\right] = \sum_{m=0}^{M} \alpha_m \frac{T_{finish}\left(M\right) + S_m}{2} \qquad (1)$$

Here $T_{finish}\left(M\right)$ and S_m are the finish time of processing the dataset and the time to start searching. These are obtained in [2]. Intuitively, this equation holds that the average record search time is the weighted sum of the mid point of each segment weighted by the size of the segment.

III. Equivalent Expected Time of Searching for Multiple records

The solution of the expected time of searching for multiple records quickly becomes unmanageable since the last record to appear in the dataset is not always the last record to be detected. This occurs because the processors shift through the data concurrently changing the order in which detection is made. However, in the case of a single processor, the last record is detected last. Furthermore, since this is a linear model, the expected time of finding *all* of the records can be obtained by applying the mean of the last record position.

A multiprocessor will be modeled as a single equivalent processor. For the case of a single equivalent processor denote for convenience the transformation function as $\mathbf{Y_L} = \mathbf{g_{eq}}\left(\mathbf{X_L}\right)$. Assuming that L records are distributed uniformly in the normalized data, the expected value of the last record location, $\mathrm{E}\left[\mathbf{X}_L\right]$ is $2\mu\frac{L}{L+1}$ [3]. The explicit form of expected time of searching for multiple records is represented as:

$$\begin{aligned} \mathrm{E}[\mathbf{Y}_L] \quad &= \quad 2\frac{L}{L+1}\left[T_{finish}\left(M\right) - \sum_{m=0}^{M}\alpha_m \frac{T_{finish}\left(M\right) + S_m}{2}\right] \\ &\quad + \left[\sum_{m=0}^{M}\alpha_m\left(T_{finish}\left(M\right) + S_m\right) - T_{finish}\left(M\right)\right] \end{aligned}$$

IV. Conclusion

It should be noted that networks as general as a hypercube network can be analyzed using a spanning tree network load distribution model. Furthermore a general tree network which has more than one level can be transformed into a single level tree network by collapsing parent processors with children processors into equivalent processors starting from the terminal nodes in order to solve for its expected time. The methodology of this work [2] is then applicable. The use of divisible load theory to solve for expected record search time is noteworthy as it leads to a tractable analysis and elegant results. The approach taken here can be used for a variety of interconnection topologies and load distribution schedules.

References

[1] V. Bharadwaj, D. Ghose, V. Mani and T.G. Robertazzi "*Scheduling Divisible Loads in Parallel and Distributed Systems,*" IEEE Computer Society Press, Los Alamitos, CA, 1996.

[2] K. Ko and T.G. Robertazzi, "*Record Search Time Evaluation Using Divisible Load Analysis,*" SUNY at Stony Brook College of Engineering and Applied Science Technical Report 765, Sep. 8, 1998.

[3] H. L. Harter, "*Ordering Statistics and Their Use in Estimation and Testing,*" Vols. 1 and 2, Superintendent of Documents, U.S. Government Printing Office, 1969.