LOAD BALANCING INSTABILITIES DUE TO TIME DELAYS IN PARALLEL COMPUTATIONS

Chaouki Abdallah [*] **J. Douglas Birdwell** [**]
John Chiasson [**] **Victor Chupryna** [**] **Zhong Tang** [**]
Tsewei Wang [***]

[*] *ECE Dept, University of NewMexico, Alburquerque NM 87131-1356, USA, chaouki@eece.unm.edu*
[**] *ECE Dept, University of Tennessee, Knoxville TN 37996, USA, birdwell@utk.edu, chiasson@utk.edu, tang@hickory.engr.utk.edu*
[***] *ChEDept, University of Tennessee, Knoxville TN 37996, USA, twang@utk.edu*

Abstract: A deterministic dynamic linear time-delay model is presented to model load balancing in a cluster of nodes used for parallel computations. The model is analyzed for stability in terms of the delays in the transfer of information between nodes and the gains in the load balancing algorithm.

Keywords: Time Delay, Stability, Load Balancing, Parallel Computation, Cluster Computing

1. INTRODUCTION

Parallel computer architectures utilize a set of computational elements (CE) to achieve performance that is not attainable on a single processor, or CE, computer. A common architecture is the cluster of otherwise independent computers communicating through a shared network. To make use of parallel computing resources, problems must be broken down into smaller units that can be solved individually by each CE while exchanging information with CEs solving other problems.

The Federal Bureau of Investigation (FBI) National DNA Indexing System (NDIS) and Combined DNA Indexing System (CODIS) software are candidates for parallelization. New methods developed by Wang et al (Wang, 2001)(Wang, 1999)(Birdwell, 2000)(Birdwell, 1999)(Birdwell, 2001) lead naturally to a parallel decomposition of the DNA database search problem while providing orders of magnitude improvements in performance over the current release of the CODIS software. The projected growth of the NDIS database and in the demand for searches of the database necessitates migration to a parallel computing platform.

Effective utilization of a parallel computer architecture requires the computational load to be distributed more or less evenly over the available CEs. The qualifier "more or less" is used because the communications required to distribute the load consumes both computational resources and network bandwidth. A point of diminishing returns exists.

Distribution of computational load across available resources is referred to as the *load balancing* problem in the literature. Various taxonomies of load balancing algorithms exist. Direct methods examine the global distribution of computa-

tional load and assign portions of the workload to resources before processing begins. Iterative methods examine the progress of the computation and the expected utilization of resources, and adjust the workload assignments periodically as computation progresses. Assignment may be either deterministic, as with the dimension exchange/diffusion (Corradi and Zambonelli, 1999) and gradient methods, stochastic, or optimization based. A comparison of several deterministic methods is provided by Willeback-LeMain and Reeves (Willebeek-LeMair and Reeves, 1993).

To adequately model load balancing problems, several features of the parallel computation environment should be captured: (1) The workload awaiting processing at each CE; (2) the relative performances of the CEs; (3) the computational requirements of each workload component; (4) the delays and bandwidth constraints of CEs and network components involved in the exchange of workloads, and (5) the delays imposed by CEs and the network on the exchange of measurements. A queuing theory (Kleinrock, 1975) approach is well-suited to the modeling requirements and has been used in the literature by Spies (Spies, 1996) and others. However, whereas Spies assumes a homogeneous network of CEs and models the queues in detail, the present work generalizes queue length to an expected waiting time, normalizing to account for differences among CEs, and aggregates the behavior of each queue using a continuous state model. The present work focuses upon the effects of delays in the exchange of information among CEs, and the constraints these effects impose on the design of a load balancing strategy.

2. MODEL

The mathematical model of a given computing node is given by

$$\begin{aligned} \frac{dx_i(t)}{dt} &= \lambda_i - \mu_i + u_i(t) - \sum_{j\neq i} p_{ij}u_j(t) \\ y_i(t) &= x_i(t) - \frac{\sum_{j=1}^{n} x_j(t-\tau_{ij})}{n} \\ u_i(t) &= -K_i y_i(t-h_i) \end{aligned} \tag{1}$$

where: $x_i(t)$ is the expected waiting time experienced by a task inserted into the queue of the i^{th} node, q_i is the number of tasks in the i^{th} node, t_{p_i} is average time needed to process a task on the i^{th} node ($x_i(t) = q_i t_{p_i}$), λ_i is the rate of generation of waiting time on the i^{th} node caused by the addition of tasks (rate of increase in x_i), μ_i is the rate of reduction in waiting time caused by the service of tasks at the i^{th} node ($\mu_i \equiv (1 \times t_{p_i})/t_{p_i} = 1$ for all i), $u_i(t)$ is the rate of removal (transfer) of the tasks from node i at time t by load balancing at node i, p_{ij} is the fraction of $u_i(t)$ that node j allocates to node i at time t; $\sum_{i=1}^{n} p_{ij} = 1$, ($p_{jj} = 0$), $p_{ij}u_j(t-h_{ij})$ is the rate of removal (transfer) of tasks at time t from node j by (to) node i, h_{ij} is the time delay for task transfer from node j to node i. ($h_{ii} = 0$), τ_{ij} is the time delay for communicating the node waiting time x_j to node i ($\tau_{ii} = 0$), and n is the number of nodes. All rates are in units of the rate of change of expected waiting time (time/time, which is dimensionless).

Here $u_i(t) < 0$ means tasks are being sent to other nodes while $u_i(t) > 0$ means the i^{th} node is receiving tasks from other nodes. A delay is experienced by transmitted tasks before they are received at the other node. The control law $u_i(t) = -K_i y_i(t-h_i)$ states that if the i^{th} node output $x_i(t)$ is above the local average $\left(\sum_{j=1}^{n} x_j(t-\tau_{ij})\right)/n$ then it sends data to the other nodes, while if it is less than the local average, it accepts data from the other nodes. Here h_i is the delay in sending the tasks $-K_i y$ to the other nodes.

Often, the p_{ij} are functions of the state x_i so as to send a higher fraction of the data to those nodes that have less tasks. However, this is left out in this model to retain linearity of the system so that a stability analysis can be carried out.

3. STABILITY ANALYSIS

To study the stability of the model, three nodes ($n = 3$) are considered with $K_1 = K_2 = K_3 = K$, $p_{ij} = 1/2$, for all $i \neq j$, $p_{ii} = 0$, $\tau_{ij} = h$ for $i \neq j$, $\tau_{ii} = 0$, $h_i = 2h$ for all $i = 1, 2, 3$. The transfer function from the inputs $d_1 = \lambda_1 - \mu_1$, $d_2 = \lambda_2 - \mu_2$, and $d_3 = \lambda_3 - \mu_3$ to the output $y_1(s) \triangleq x_1(s) - \left(x_1(s) + e^{-hs}x_2(s) + e^{-hs}x_3(s)\right)/3$ is ($z = e^{-hs}$)

$$\begin{aligned} &y_1(s) = \\ &\frac{-\frac{1}{18}(6s + K(2z^2 - z^3 - z^4))(2s + K(2z^2 + z^3)}{-\frac{1}{4}s(2s + K(2z^2 + z^3)^2} d_1 \\ &+ \frac{\frac{1}{18}z\left(3s + K(-2z + z^2 + z^3)\right)(2s + K(2z^2 + z^3)}{-\frac{1}{4}s(2s + K(2z^2 + z^3)^2} \times \\ &(d_2 + d_3) \\ &= \frac{2}{9}\frac{6s + K(2z^2 - z^3 - z^4)}{s(2s + K(2z^2 + z^3)} d_1 \\ &- \frac{2}{9}\frac{z\left(3s + K(-2z + z^2 + z^3)\right)}{s(2s + K(2z^2 + z^3))}(d_2 + d_3) \\ &= \frac{2}{9}\frac{b_1(s,z)}{sa(s,z)} d_1 - \frac{2}{9}\frac{zb_2(s,z)}{sa(s,z)}(d_2 + d_3) \end{aligned}$$

where $b_1(s,z) = 6s + K(2z^2 - z^3 - z^4)$, $b_2(s,z) = 3s + K(-2z + z^2 + z^3)$ and $a(s,z) = 2s + K(2z^2 + z^3)$.

The stability of this time delay system is analyzed using the techniques developed in (Bellman and

Cooke, 1963)(Chiasson and Lee, 1985)(Hertz and Zeheb, 1984)(Kamen, 1982)(Chiasson, 1988)(Chiasson and Abdallah, 2001) (see especially (Chiasson, 1988)(Chiasson and Abdallah, 2001)). The problem is broken into the three parts where it is first shown that $a(s,e^{-hs})$ for stable for $0 \le h < \frac{2}{K}\frac{0.674889}{2.85}$. Then it is shown that $b_1(s,e^{-hs})/s$ and $b_2(s,e^{-hs})/s$ are both stable independent of delay.

3.1 Stability of $a(s,e^{-hs})$

First $a(s,z)$ is considered. With $K>0$, the polynomial $a(s,z)$ is stable for $h=0$ since $a(s,1) = 2s+3K$. Next, define

$$a(s,z) = 2s + K(2z^2+z^3)$$
$$\tilde{a}(s,z) \triangleq z^3 a(-s,1/z) = z^3(-2s) + K(2z+1)$$

The idea here (see (Chiasson, 1988)(Chiasson and Abdallah, 2001)) is to simply note that the polynomial is stable for $h=0$ and then to increase h until there are zeros of the polynomial on the $j\omega$ axis. At this point, there are zeros of the form $s = j\omega, z = e^{-j\omega h}$ which must satisfy $a(s,z) = \tilde{a}(s,z) = 0$ as $-j\omega$ and $1/e^{-j\omega h}$ are simply the conjugates of $s = j\omega, z = e^{-j\omega h}$, respectively. To find the first value of h that results in the system being unstable, the common zeros of $\{a(s,z), \tilde{a}(s,z)\}$ on the $j\omega$ axis are found. Specifically, the variable s is eliminated from $a(s,z)$ and $\tilde{a}(s,z)$ resulting in

$$\begin{aligned} R(z) &= z^6 + 2z^5 + 2z + 1 \\ &= (z^2+2z+1)(z^4+z^3-2z^2+z+1) = 0. \end{aligned}$$

These roots are

$$\begin{aligned} z_{1,2} &= -\frac{1}{2} \pm j\frac{\sqrt{3}}{2} = e^{j2\pi/3}, e^{j4\pi/3} \\ z_{3,4} &= \frac{-1+\sqrt{17}}{4} \pm j\frac{1}{\sqrt{2}}\sqrt{\frac{-1+\sqrt{17}}{4}} \\ &= 0.780776 \pm j0.624811 \\ z_{5,6} &= \frac{-1-\sqrt{17}}{4} \pm \frac{1}{\sqrt{2}}\sqrt{\frac{1+\sqrt{17}}{4}} \end{aligned}$$

Only the first four roots $z_{1,2}$,$z_{3,4}$ have magnitude one and the corresponding values of s are

$$s_{1,2} = -\frac{K}{2}(2z^2+z^3)\Big|_{z=-\frac{1}{2}\pm j\frac{\sqrt{3}}{2}} = -\frac{K}{2}\left(\frac{1}{2} \mp j\frac{\sqrt{3}}{2}\right)$$
$$s_{3,4} = -\frac{K}{2}(2z^2+z^3)\Big|_{0.780776\pm j0.624811} = \mp\frac{K}{2}2.85j$$

As $\mathrm{Re}(s_{1,2}) \neq 0$, these do not correspond to zeros of $a(s,e^{-hs})$ on the $j\omega$ axis. Setting $z = e^{-sh}$ with $s = \mp\frac{K}{2}2.85j$, $z = 0.780776 \pm j0.624811 = e^{\pm j0.674889}$ gives

$$e^{\pm j0.674889} = e^{\pm\frac{K}{2}2.85jh}$$
$$\Longrightarrow h = \frac{2}{K}\frac{0.674889}{2.85}$$

That is, for a given K, this gives the smallest positive value of the delay for which $a(s,e^{-hs})$ has a zero in the right-half plane (specifically, on the $j\omega$ axis). Or, in terms of the gain K, $a(s,e^{-hs})$ has no zeros in the right-half plane for

$$K < K_{\max} \triangleq \frac{2}{h}\frac{0.674889}{2.85}$$

and further has zeros on the $j\omega$ axis for $K = K_{\max}$.

3.2 Stability of $b_1(s,e^{-hs})/s$ and $b_2(s,e^{-hs})/s$

Consider next transfer function

$$\frac{b_1(s,e^{-hs})}{s} = \frac{6s + K(2e^{-2hs} - e^{-3hs} - e^{-4hs})}{s}$$

The denominator has a single root at $s=0$. However, this is not a pole since

$$\begin{aligned} \frac{b_1(s,e^{-hs})}{s}\Big|_{s=0} &= \lim_{s\to 0}\frac{b_1(s,e^{-hs})}{s} \\ &= \lim_{s\to 0}\frac{6s + K(2e^{-2hs} - e^{-3hs} - e^{-4hs})}{s} \\ &= \lim_{s\to 0}\frac{6s + 3Khs}{s} \\ &= 6 + 3K \neq \infty \end{aligned}$$

Similarly, for the transfer function

$$\frac{b_2(s,e^{-hs})}{s} = \frac{3s + K(-2e^{-hs} + e^{-2hs} + e^{-3hs})}{s}$$

the denominator has a single root at $s=0$. As before, this is not a pole since

$$\begin{aligned} \frac{b_2(s,e^{-hs})}{s}\Big|_{s=0} &= \lim_{s\to 0}\frac{b_2(s,e^{-hs})}{s} \\ &= \lim_{s\to 0}\frac{3s + K(-2e^{-hs} + e^{-3hs} + e^{-3hs})}{s} \\ &= \lim_{s\to 0}\frac{3s - 4Khs}{s} \\ &= 3 - 4Kh \neq \infty \end{aligned}$$

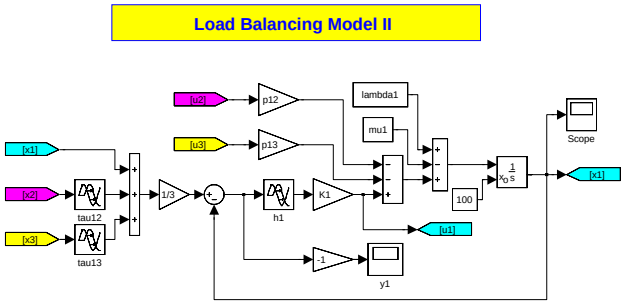
Consequently, the transfer functions $b_1(s,e^{-hs})/s$, $b_2(s,e^{-hs})/s$ are both stable independent of delay.

4. SIMULATIONS

Experimental procedures to determine the delay values are given in the thesis (Dasgupta, 2001*a*) and summarized in the paper (Dasgupta, 2001*b*). These give representative values for a Fast Ethernet network with three nodes of $\tau_{ij} = 400\ \mu$sec for $i \neq j, \tau_{ii} = 0$, and so with $h_i = 2 \times 400\ \mu$sec, the maximum value for the gain is

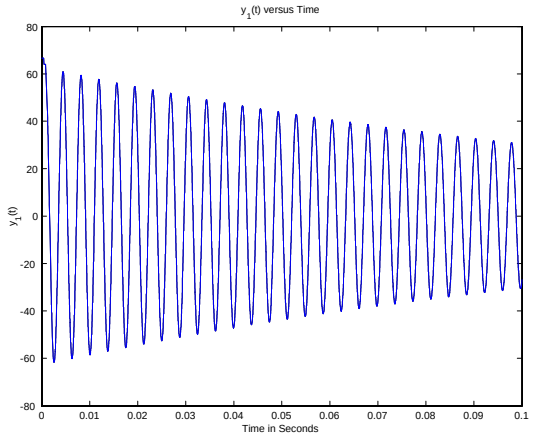
$$K_{\max} = \frac{2}{h}\frac{0.674889}{2.85} = \frac{2}{400\times 10^{-6}}\frac{0.674889}{2.85} = 1184$$

The simulation was performed with three nodes ($n=3$), $K_1 = K_2 = K_3 = K$, $p_{ij} = 1/2$, for all i,j, $\tau_{ij} = h$ for $i \neq j$, $\tau_{ii} = 0$, $h_i = 2h$ for $i =$

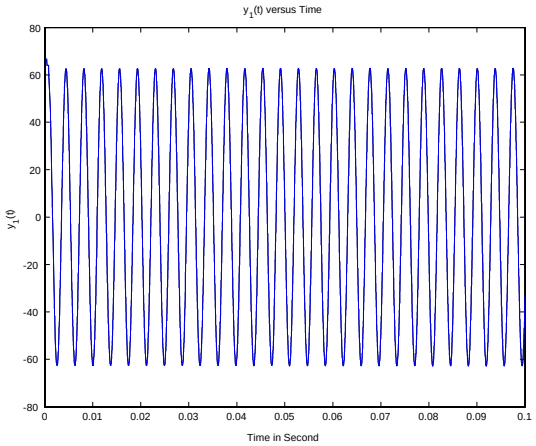


$1, 2, 3$ and $h = 400$ μsec. The inputs were set as $\lambda_1 = 2\mu_1, \lambda_2 = 0, \lambda_3 = 0$ and the initial conditions were $x_1(0) = 100$, $x_2(0) = 5$ and $x_3(0) = 3$. The figure below is a block diagram of one node of the system where $K_1 = K$, $\tau_{12} = \tau_{13} = 400$ μsec and $h_1 = 800$ μsec.

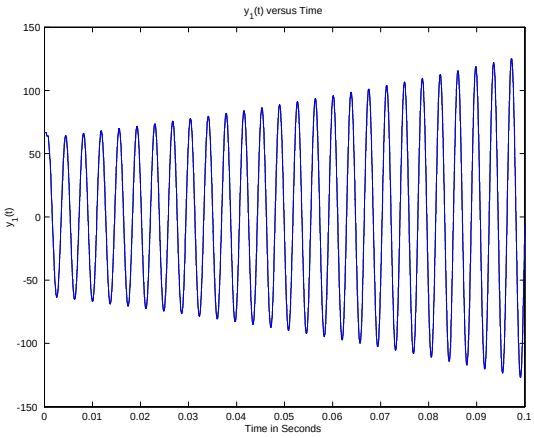
The figures below are plots of $y_1(t)$ from the simulation for three runs where the gain is set as $K = 0.99K_{\max}, K = K_{\max}$, and $K = 1.01K_{\max}$, respectively.



Plot of $y_1(t)$ versus time for $K = 0.99K_{\max}$



Plot of $y_1(t)$ versus time for $K = K_{\max}$



Plot of $y_1(t)$ versus time for $K = 1.01K_{\max}$

5. CONCLUSIONS

In this work, a load balancing algorithm was modeled as a linear time-delay system. Under the assumption of symmetric nodes and controllers (all intercommunication delays are identical and the controller gains identical) a systematic procedure was presented to determine the stability of the system by an explicit relationship between the delay values and the control gain. The delays create a limit on the size of the controller gains in order to ensure stability.

6. ACKNOWLEDGEMENTS

The work of J.D. Birdwell, V. Chupryna, Z. Tang, and T.W. Wang was supported by U.S. Department of Justice, Federal Bureau of Investigation under contract J-FBI-98-083. The work of C.T. Abdallah was supported in part by the National Science Foundation through the grant INT-9818312. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government.

REFERENCES

Bellman, R. and K.L. Cooke (1963). *Differential-Difference Equations*. New York: Academic.

Birdwell, J.D., R. D. Horn D. J. Icove T. W. Wang P. Yadav S. Niezgoda (1999). A hierarchical database design and search method for codis. In: *Tenth International Symposium on Human Identification*. Orlando, FL.

Birdwell, J.D., T-W. Wang M. Rader (2001). The university of tennessee's new search engine for codis. In: *6th CODIS Users Conference*. Arlington, VA.

Birdwell, J.D., T. W. Wang R. D. Horn P. Yadav D. J. Icove (2000). Method of indexed storage and retrieval of multidimensional information. U. S. Patent Application 09/671,304.

Chiasson, J. (1988). A method for computing the interval of delay values for which a differential-delay system. *IEEE Transactions on Automatic Control* **33**(12), 1176–1178.

Chiasson, J. and C.T. Abdallah (2001). A test for robust stability of time delay systems. Sante Fe, NM.

Chiasson, J.N., S.D. Brierley and E.B. Lee (1985). A simplified derivation of the zeheb-walach 2-d stability test with applications to time-delay systems. *IEEE Transactions on Automatic Control*.

Corradi, A., L. Leonardi and F. Zambonelli (1999). Diffusive load-balancing polices for dynamic applications. *IEEE Concurrency* **22**(31), 979–993.

Dasgupta, P. (2001*a*). *Performance Evaluation of Fast Ethernet, ATM and Myrinet under PVM, MS Thesis*. University of Tennesse.

Dasgupta, P., J. D. Birdwell T. W. Wang (2001*b*). Timing and congestion studies under pvm. In: *Tenth SIAM Conference on Parallel Processing for Scientific Computation*. Portsmouth, VA.

Hertz, D., E.I. Jury and E. Zeheb (1984). Simplified analytic stability test for systems with commensurate time delays. *IEE Proceedings*. part D.

Kamen, E.W. (1982). Linear systems with commensurate time delays: Stability and stabilization independent of delay. *IEEE Transactions on Automatic Control* pp. 367–375.

Kleinrock, L. (1975). *Queuing Systems Vol I : Theory*. John Wiley & Sons. New York.

Spies, F. (1996). Modeling of optimal load balancing strategy using queuing theory. *Microprocessors and Microprogramming* **41**, 555–570.

Wang, T.W., J. D. Birdwell P. Yadav D. J. Icove S. Niezgoda S. Jones (1999). Natural clustering of dna/str profiles. In: *Tenth International Symposium on Human Identification*. Orlando, FL.

Wang, T.W., P. Yadav J. D. Birdwell (2001). Method of index storage and retrieval of dna profiles stored in a large database. In: *American Institute of Chemical Engineering Annual Conference*. Reno Nevada.

Willebeek-LeMair, M.H. and A.P. Reeves (1993). Strategies for dynamic load balancing on highly parallel computers. *IEEE Transactions on Parallel and Distributed Systems* **4**(9), 979–993.