Short Notes

Message Complexity of the Tree Quorum Algorithm

Shyan-Ming Yuan and Her-Kun Chang

***Abstract*—The tree quorum algorithm (TQA) uses a tree structure to generate intersecting (tree) quorums for distributed mutual exclusion. This paper analyzes the number of messages required to acquire a quorum in TQA. Let i be the depth of the complete binary tree used in TQA, and let M_i be the number of messages required to acquire a quorum or to determine that no quorum is accessible. We discuss M_i as a function of i and p, where $p\left(\frac{1}{2} < p < 1\right)$ is the probability that each site is operational. Let C_i denote the average number of sites in the quorum that TQA finds. The analysis shows that, although both M_i and C_i increase without bound as i increases, M_i/C_i approaches to $\frac{1+p}{p}$ as i increases. According to the result, an approximate close form for M_i is derived.**

***Index Terms*—Distributed mutual exclusion, tree quorum algorithm, quorum size, message complexity.**

I. Introduction

A distributed system consists of a set of sites which are loosely coupled by a computer network. One advantage of distributed systems is resource sharing. That is the resources in a distributed system can be shared among the sites in the system. Examples of sharable resources are memory, peripheral, CPU, clock, etc. The sites in a distributed system may issue requests to a shared resource at arbitrary times. When two or more sites try to access the same resource at the same time, a conflict occurs. A mechanism is required to synchronize conflicting requests so that at most one site is allowed to access the resource at a time. This problem is known as distributed mutual exclusion [1], [2], [3], [4], [5]. A survey of various algorithms for mutual exclusion can be found in [4] and a simple taxonomy for distributed mutual exclusion algorithms was reported in [5].

A central controller can be used to control mutually exclusive accesses to a shared resource. All requests for the resource are sent to the controller and scheduled by the controller. Using a central controller is simple and easy to implement. However, the controller is vulnerable to site failure. When the controller fails, no access to the resource is allowed, i.e., the entire system is *halted*. It is desirable to reduce the probability that the system is halted by using more than one site to participate in the decision making. For example, majority consensus [6] can be used to achieve mutual exclusion wherein a site is allowed to access the resource if it can get permissions from a majority of sites.

Quorum consensus is a generalization of majority consensus. Let U be the set of sites in a system. A *quorum* Q is a subset of U and each access is allowed to perform if it can get permissions from all sites in a quorum. To synchronize the accesses in a mutually exclusive way, the quorums must satisfy the following property:

For each pair of quorums Q_1 and Q_2, $Q_1 \cap Q_2 \neq \varnothing$.

Mutual exclusion is ensured by requiring the accesses to get permissions from intersecting quorums. Since the quorums intersect with each other, it is impossible that two accesses can get permissions from two quorums at the same time.

The communication cost of a quorum consensus algorithm can be measured by the following metrics:

- *message complexity*-expected number of messages that the algorithm uses to acquire a quorum or to determine no quorum is accessible.
- *quorum size*-expected number of sites in the quorum that the algorithm finds.

So far as we know, the communication cost of each quorum consensus algorithm proposed in the literature is estimated by the quorum size. The quorum size in majority consensus is $\left\lceil \frac{N+1}{2} \right\rceil$. The tree quorum algorithm (TQA) uses a tree structure to generate tree quorums [1]. The size of the tree quorums varies from log N to $\left\lceil \frac{N+1}{2} \right\rceil$.

In general, the communication cost can be measured by message complexity more precisely than by quorum size. It was assumed in [1] that the number of messages required to construct a quorum is directly proportional to the size of the quorums. That is (in terms of this paper) message complexity is proportional to quorum size. The assumption motivates us to study the relationship between message complexity and quorum size. Let M_i be the message complexity of an i level (complete binary) tree, and let C_i be the quorum size of the i level tree. We discuss M_i as a function of i and p, where p $\left(\frac{1}{2} < p < 1\right)$ is the probability that each site is operational. To verify the assumption, an asymptotic analysis of the ratio of message complexity to quorum size, $R_i = M_i/C_i$, is presented. The analysis shows that, although both M_i and C_i increase without bound as i increases, M_i/C_i approaches to $\frac{1+p}{p}$ as i increases.

Although both M_i and C_i can be computed by recurrence equations, C_i has a close form but M_i has no close form. An important implication of the analytic result is:

As i increases, $M_i/C_i \approx \frac{1+p}{p}$ and $M_i \approx \frac{1+p}{p} C_i$. That is, an approximate close form for M_i can be derived.

The remainder of the paper is organized as follows. Section II briefly reviews TQA. In Section III, message complexity of TQA is analyzed and an asymptotic analysis of the ratio of message complexity to quorum size is presented. Some concluding remarks are given in the final section.

II. Tree Quorum Algorithm

The model in the analysis is described as follows. The sites are assumed to be fully connected by perfect links. When a request is sent to a site, a reply is sent if the site is operational. If the site has failed, no reply is sent.

The TQA uses a tree structure to generate (tree) quorums. The analysis in this paper consider only complete binary trees. For a binary tree, a tree quorum (recursively) consists of

1) the root and a tree quorum of the left or right subtree, or

Manuscript received Mar. 2, 1994; revised July 15, 1994.

S.-M. Yuan and H.-K. Chang are with the Department of Computer and Information Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30050, Taiwan; e-mal smyuan@cis.nctu.edu.tw.

IEEECS Log Number D95031.

1045-9219/95$04.00 © 1995 IEEE

2) a tree quorum of the left subtree and a tree quorum of the right subtree.

It was shown in [1] that there is a nonnull intersection between each pair of (tree) quorums. Thus mutual exclusion is ensured by requiring that each access to get permissions from any quorum of sites.

The sites are either *operational* or *failed*. The state of a site (operational or failed) is independent to the others. The probability that a site is operational, is referred to as the availability of the site. The availability of a tree is the probability that a quorum can be acquired from the tree.

The analysis uses the following notations

- p : availability of a single site, $1/2 < p < 1$,
- A_i : availability of an i level tree, $i \geq 0$,
- C_i: quorum size of an i level tree, $i \geq 0$,
- M_i : message complexity of an i level tree, $i \geq 0$,
- R_i : M_i / C_i, $i \geq 0$.

The availability of a tree is the probability that a quorum can be acquired from the tree. Thus the availability of a binary tree is the probability that

1) the root is operational and a tree quorum of the left or right subtree is available, or
2) the root is failed, a tree quorum of the left subtree is available and a tree quorum of the right subtree is available.

Formally, $A_0 = p$ and A_i, $1 \leq i \leq k$, is given as

$$\begin{aligned} A_i &= p\left(1-\left(1-A_{i-1}\right)^2\right)+(1-p)A_{i-1}^2 \\ &= A_{i-1}^2 + 2pA_{i-1}\left(1-A_{i-1}\right) \end{aligned} \tag{1}$$

If

$$p \leq \tfrac{1}{2},\ A_i - A_{i-1} = (2p-1)A_{i-1}\left(1-A_{i-1}\right) \leq 0, \text{ i.e.}, A_i \leq A_{i-1},$$

for all $i \geq 1$. If $p = 1$, $A_i = 1$, for all $i \geq 0$. So the analysis considers only the case that $\frac{1}{2} < p < 1$.

The quorum size of a 0 level tree (i.e., a tree consists of only one site) is 1, i.e., $C_0 = 1$. Consider constructing a tree quorum at level i, $i \geq 1$. If the root is operational and thus can be included in the quorum, the quorum size is $1 + C_{i-1}$; otherwise, the quorum size is $2C_{i-1}$. Thus, for $i \geq 1$, C_i can be computed by the following recurrence:

$$\begin{aligned} C_i &= p\left(1+C_{i-1}\right)+(1-p)2C_{i-1} \\ &= (2-p)C_{i-1}+p \end{aligned} \tag{2}$$

III. Analysis of TQA

In this section, message complexity of TQA is analyzed. An asymptotic analysis of the ratio of message complexity to quorum size, R_i, is presented. It is shown that R_i converges to $\frac{1+p}{p}$ as i increases.

A. Message Complexity

The recursive definition of the tree quorums in the previous section also implies a quorum construction algorithm. That is, if the root is operational, then the construction algorithm tries to construct a (tree) quorum from left or right subtree; otherwise, it must construct quorums from both left and right subtrees. In other words, the quorum construction algorithm first visits the root and then traverses the left and/or right subtrees (in some specified order or randomly).

First consider constructing a tree quorum from a 0 level tree, (i.e., a tree consists only one site).

1) If the site is operational, two messages are transmitted–one request and one reply.
2) Otherwise, only one message is sent–no reply.

Thus,

$$\begin{aligned} M_0 &= 2p + (1-p) \\ &= 1 + p \end{aligned} \tag{3}$$

Consider constructing an i level tree quorum, $i \geq 1$. Without loss of generality, we assume that the quorum construction algorithm tries to acquire a quorum (recursively) by the order: root, left subtree and right subtree. If the root is operational, two messages are transmitted; otherwise, only one message is sent. Thus, in average, $1 + p$ messages are sent. The messages required to traverse the subtrees are described below:

1) if the root is operational and a quorum of the left subtree is available-only messages for traversing the left subtree are sent, i.e., M_{i-1} messages are needed;
2) if the root is failed and no quorum of the left subtree is available-since it is impossible to acquire a quorum, only messages for traversing the left subtree are sent, i.e., M_{i-1} messages are needed;
3) otherwise–messages are required to traverse both left and right subtrees, i.e., 2 M_{i-1} messages are required.

Formally, for i ≥ 1,

$$\begin{aligned} M_i &= (1+p) \\ &\quad +\left(pA_{i-1}+(1-p)\left(1-A_{i-1}\right)\right)M_{i-1} \\ &\quad +\left(p\left(1-A_{i-1}\right)+(1-p)A_{i-1}\right)2M_{i-1} \\ &= \left(1+p+A_{i-1}-2pA_{i-1}\right)M_{i-1}+(1+p) \end{aligned} \tag{4}$$

B. Asymptotic Analysis of R_i

Lemma 1. (Lemma 2 of [7]) *If* $0 < ax_i < 1$, *for all i, then*

$$\left(1+ax_1\right)\left(1+ax_2\right)\cdots\left(1+ax_n\right) < e^{a\sum x_i}$$

Lemma 2. (Lemma 3 of [7]) *For TQA*, $\frac{1}{2} < p < 1$, A_i *has the following properties*:

1) $1 - A_i \leq (1-p)(1+p-2p^2)^i$, *for all* $i \geq 0$.

2) $\left(1-A_i\right)+\left(1-A_{i+1}\right)+\cdots+\left(1+A_{i+m}\right) < \frac{1-p}{p(2p-1)}\left(1+p-2p^2\right)^i$, *for all i*, $m \geq 0$.

Lemma 3. (Lemma 4 of [7]) *For TQA*, C_i *has the following properties*:

1) $C_i = \dfrac{(2-p)^i - p}{1-p} > (2-p)^i$, for all $i \geq 0$.

2) $\frac{1}{C_i}+\frac{1}{C_{i+1}}+\cdots+\frac{1}{C_{i+m}} < \frac{2-p}{1-p}\frac{1}{(2-p)^i}$, for all i, $m \geq 0$.

Lemma 4. *For TQA*, $i \geq 0$,

$$R_i \geq 1 + p$$

Proof. The proof is shown in the appendix.

Lemma 5. *For TQA*, $\frac{1}{2} < p < 1$,

$$R_i < (1+p)e^{\frac{1+p-p^2}{p(2-p)}}$$

PROOF. From (2) and (4), we have, for $i \geq 1$,

$$\begin{aligned}
R_i &= \tfrac{M_i}{C_i} \\
&= \frac{(1+p+A_{i-1}-2pA_{i-1})M_{i-1}+(1+p)}{(2-p)C_{i-1}+p} \\
&= \left(\frac{(1+p+A_{i-1}-2pA_{i-1})C_{i-1}+\frac{1+p}{R_{i-1}}}{(2-p)C_{i-1}+p}\right)R_{i-1} \\
&= \left(1+\frac{(2p-1)(1-A_{i-1})C_{i-1}}{(2-p)C_{i-1}+p}+\frac{\frac{1+p}{R_{i-1}}-p}{(2-p)C_{i-1}+p}\right)R_{i-1}
\end{aligned}$$

By Lemma 4, $R_{i-1} \geq 1+p$, we obtain

$$\begin{aligned}
R_i &\leq \left(1+\frac{(2p-1)(1-A_{i-1})C_{i-1}}{(2-p)C_{i-1}+p}+\frac{1-p}{(2-p)C_{i-1}+p}\right)R_{i-1} \\
&< \left(1+\frac{(2p-1)(1-A_{i-1})}{2-p}+\frac{1-p}{C_i}\right)R_{i-1}
\end{aligned}$$

By iteration,

$$R_i < (1+p)\left(1+\frac{(2p-1)(1-A_0)}{2-p}+\frac{1-p}{c_1}\right)\cdots\left(1+\frac{(2p-1)(1-A_{i-1})}{2-p}+\frac{1-p}{C_1}\right)$$

According to Lemmas 1, 2, and 3,

$$\begin{aligned}
R_i &< (1+p)e^{\left(\frac{2p-1}{2-p}((1-A_0)+\cdots+(1-A_{i-1}))+(1-p)\left(\frac{1}{C_1}+\cdots+\frac{1}{Ci}\right)\right)} \\
&< (1+p)e^{\left(\frac{2p-1}{2-p}\frac{1-p}{p(2p-1)}+(1-p)\left(\frac{2-p}{1-p}\right)\frac{1}{2-p}\right)} \\
&= (1+p)e^{\frac{1+p-p^2}{p(2-p)}}
\end{aligned}$$

□

LEMMA 6. *If* $\frac{1}{2} < p < 1, \left|R_{i+m}-R_i\right| < ax^i+by^i$, *for all* $i \geq 0$, $m \geq 1$, *where*

$$a = \frac{1-p^2}{p(2-p)}e^{\frac{1+p-p^2}{p(2-p)}}$$

$$x = 1+p-2p^2$$

$$b = 1+p$$

$$y = \frac{1}{2-p}$$

PROOF. Let $\delta_{i+1} = R_{i+1} - R_i$, then

$$\begin{aligned}
\delta_{i+1} &= \frac{(1+p+A_i-2pA_i)M_i+(1+p)}{(2-p)C_i+p}-\frac{M_i}{C_i} \\
&= \frac{(2p-1)(1-A_i)M_i+(1+p)-p\frac{M_i}{C_i}}{(2-p)C_i+p}
\end{aligned}$$

By Lemma 4, $R_i = M_i/C_i \geq 1+p$, we obtain

$$\begin{aligned}
\delta_{i+1} &\leq \frac{(2p-1)(1-A_i)M_i}{(2-p)C_i+p}+\frac{(1+p)-p(1+p)}{C_{i+1}} \\
&< \frac{(2p-1)(1-A_i)M_i}{(2-p)C_i}+\frac{(1-p^2)}{C_{i+1}} \\
&= \frac{(2p-1)(1-A_i)}{(2-p)}R_i+\frac{1-p^2}{C_{i+1}}
\end{aligned}$$

According to Lemma 5

$$\delta_{i+1} < \frac{(2p-1)(1-A_i)}{(2-p)}(1+p)e^{\frac{1+p-p^2}{p(2-p)}}+\frac{1-p^2}{C_{i+1}}$$

Therefore,

$$\begin{aligned}
\left|R_{i+m}-R_i\right| &= \left|\delta_{i+m}+\delta_{i+m-1}+\cdots+\delta_{i+1}\right| \\
&< \frac{2p-1}{(2-p)}(1+p)e^{\frac{1+p-p^2}{p(2-p)}}\left((1-A_i)+\cdots+(1-A_{i+m-1})\right) \\
&\quad +\left(1-p^2\right)\left(\frac{1}{C_{i+1}}+\cdots+\frac{1}{C_{i+m}}\right)
\end{aligned}$$

According to Lemma 2 and Lemma 3

$$\begin{aligned}
\left|R_{i+m}-R_i\right| &< \frac{2p-1}{(2-p)}(1+p)e^{\frac{1+p-p^2}{p(2-p)}}\frac{1-p}{p(2p-1)}\left(1+p-2p^2\right)^i \\
&\quad +\left(1-p^2\right)\frac{2-p}{1-p}\frac{1}{(2-p)^{i+1}} \\
&= \frac{1-p^2}{p(2-p)}e^{\frac{1+p-p^2}{p(2-p)}}\left(1+p-2p^2\right)^i+(1+p)\left(\frac{1}{2-p}\right)^i \\
&= ax^i+by^i
\end{aligned}$$

where

$$\begin{aligned}
a &= \frac{1-p^2}{p(2-p)}e^{\frac{1+p-p^2}{p(2-p)}} \\
x &= 1+p-2p^2 \\
b &= 1+p \\
y &= \frac{1}{2-p}
\end{aligned}$$

□

THEOREM 1. *If* $\frac{1}{2} < p < 1$,

$$\lim_{i\to\infty} R_i = \frac{1+p}{p}$$

PROOF. Let a, x, b and y be defined as in Lemma 6. Since $p > \frac{1}{2}$, $x = (1+p-2p^2) < 1$ and $y = \frac{1}{2-p} < 1$. For any $\epsilon > 0$, there is a positive integer N such that

If $i > N$ and $m \geq 1$, then

$$|R_{i+m} - R_i| < ax^i + by^i < ax^N + by^N < \epsilon$$

Thus, the sequence $\{R_i\}$ is convergent.

Let $\lim_{i\to\infty} R_i = \lim_{i\to\infty} R_{i-1} = \gamma$, that is,

$$\frac{M_i}{C_i} = \frac{M_{i-1}}{C_{i-1}} = \gamma$$

As $i \to \infty$, $A_i \to 1$. Using (2) and (4), we obtain

$$\frac{(2-p)\gamma C_{i-1}+(1+p)}{(2-p)C_{i-1}+p} = \gamma$$

$$\gamma = \frac{1+p}{p}$$

□

C. Discussion

Giving p, the probability that each site is operational, both M_i and C_i increase without bound as i increases. According to Lemma 3, C_i, $i \geq 0$, has a close form:

$$C_i = \frac{(2-p)^i - p}{1-p} \tag{5}$$

On the other hand, M_i has no close form, though M_i can also be computed by recurrence relations. Using the analytic result in the previous subsection, an approximate close form for M_i can be derived as follows. As i increases, M_i/C_i approaches to $\frac{1+p}{p}$ and thus M_i can be approximated by $\frac{1+p}{p}C_i$. From (5), we obtain

$$M_i \approx \frac{\left((2-p)^i - p\right)(1+p)}{p(1-p)} \tag{6}$$

IV. Conclusion

The communication cost can be measured by message complexity more precisely than by quorum size. So far as we know, quorum size is used to measure the communication cost of each quorum consensus proposed in the literature. It was assumed that message complexity is directly proportional to quorum size [1].

The assumption motivates us to study the relationship between message complexity and quorum size. To verify the assumption, an asymptotic analysis of the ratio of message complexity to quorum size is presented. It is shown that the ratio converges to $\frac{1+p}{p}$, where p is the probability that each site is operational. The result implies two things:

1) Giving p, the probability that each site is operational, the message complexity is proportional to the quorum size, if the tree is sufficiently large.
2) Since the quorum size can be evaluated by a close form, an approximate close form for the message complexity can be derived.

Appendix

Proof of Lemma 4.

The proof is shown by induction.

1) **Induction base**: $R_0 = M_0/C_0 = 1 + p$.
2) **Induction hypothesis**: $R_{i-1} = M_{i-1}/C_{i-1} \geq 1 + p$, i.e., $M_{i-1} \geq (1 + p)C_{i-1}$, $i \geq 1$.
3) **Induction step**: From (2) and (4), we have, for $i \geq 1$,

$$\begin{aligned} R_i &= \frac{M_i}{C_i} \\ &= \frac{(1+p+A_{i-1}-2pA_{i-1})M_{i-1}+(1+p)}{(2-p)C_{i-1}+p} \\ &= \frac{\left((2-p)+(2p-1)(1-A_{i-1})\right)M_{i-1}+(1+p)}{(2-p)C_{i-1}+p} \\ &> \frac{(2-p)M_{i-1}+(1+p)}{(2-p)C_{i-1}+p} \\ &\geq \frac{(1+p)\left((2-p)C_{i-1}+1\right)}{(2-p)C_{i-1}+p} \\ &> 1+p \end{aligned}$$

□

Acknowledgments

The authors would like to express their gratitude to the referees for their helpful comments. This work was supported in part by National Science Council of R.O.C. under the grant NSC82-0408-E009-289.

References

[1] D. Agrawal, A. El Abbadi, "An efficient and fault-tolerant solution for distributed mutual exclusion," *ACM Trans. Computer Systems*, vol. 9, no. 1, pp. 1-20, 1991.

[2] H. Garcia-Molina, D. Barbara, "How to assign votes in a distributed system," *J. ACM*, vol. 32, no. 4, pp. 841-860, 1985.

[3] T. Ibaraki, T. Kameda, "A theory of coteries: Mutual exclusion in distributed systems," *IEEE Trans. Parallel & Distributed Systems*, vol. 4, no. 7, pp. 779-794, 1993.

[4] M. Raynal, *Algorithms for mutual exclusion*, The MIT Press, 1986.

[5] M. Singhal, "A taxonomy of distributed mutual exclusion," *J. Parallel and Distributed Computing*, vol. 15, pp. 94-101, May, 1993.

[6] R.H. Thomas, "A majority consensus approach to concurrency control for multiple copy databases," *ACM Trans. Database Systems*, vol. 4, no. 2, pp. 180-209, 1979.

[7] H.K. Chang, S.M. Yuan, "Message complexity of the tree quorum algorithm for distributed mutual exclusion," *Proc. 1994 IEEE Int'l Conf. on Distributed Computing Systems*, pp. 76-80, 1994.

Performance of Barrier Synchronization Methods in a Multiaccess Network

Shun Yan Cheung and Vaidy S. Sunderam

Abstract—**Barrier synchronization is a commonly used primitive in parallel processing. In this paper, we present different algorithms for barrier synchronization on the widely prevalent multiaccess bus network, and derive analytical performance metrics for each of the proposed schemes, which are then compared against simulation results.**

Index Terms—**Distributed computing, parallel virtual machine (PVM), barrier synchronization, multiaccess networks, performance evaluation.**

I. Introduction

Barrier synchronization is a well-known and frequently used primitive in parallel processing. A barrier is a powerful mechanism that permits synchronization among a large number of cooperating processes in a parallel program, while being straightforward in terms of programming primitive(s) as well as semantics. Informally, a barrier is a function that causes the invoking process in a parallel program to be suspended until all other processes also invoke the function, at which point all processes are allowed to continue. The simplest form of barrier synchronization assumes a fixed number of related processes in a parallel application that wish to synchronize periodically; in such situations, barriers are provided as parameter-less function calls. However, variants that allow a "quorum" of participants to satisfy the barrier, or those that permit "named" barriers, also exist.

The barrier primitive originally evolved on shared-memory multiprocessors, but are currently used widely on distributed-memory multiprocessors also. Algorithms to implement barriers, as well as studies of their performance have received substantial at-

Manuscript received July 9, 1993; revised Jan. 5, 1995.

S.-Y. Cheung and V.S. Sunderam are with the Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322; e-mail: {cheung, vss}@mathcs.emory.edu.

IEEECS Log Number D95016.

1045-9219/95$04.00 © 1995 IEEE