Paper:

MDVM System Concept, Paging Latency and Round-2 Randomized Leader Election Algorithm in SG

Susmit Bagchi*, Hafizur Rahaman, and Purnendu Das*****

*Deptt. of Information Technology, Sikkim Manipal Inst. of Technology
Majitar, Sikkim, India
E-mail: susmit-bagchi@lycos.com
**Deptt. of Information Technology, Bengal Engineering and Science University
Shibpur, Howrah, India
E-mail: rahaman_h@hotmail.com
***School of Information Technology, Bengal Engineering and Science University
Shibpur, Howrah, India

[Received October 28, 2005; accepted April 17, 2006]

The future trend in the computing paradigm is marked by mobile computing based on mobile-client/server architecture connected by wireless communication network. However, the mobile computing systems have limitations because of the resource-thin mobile clients operating on battery power. The MDVM system allows the mobile clients to utilize memory and CPU resources of Server-Groups (SG) to overcome the resource limitations of clients in order to support the high-end mobile applications such as, m-commerce and virtual organization (VO). In this paper the concept of MDVM system and the architecture of cellular network containing the SG are discussed. A round-2 randomized distributed algorithm is proposed to elect a unique leader and co-leader of the SG. The algorithm is free from any assumption about network topology, buffer space limitations and is based on dynamically elected coordinators eliminating single point of failure. The algorithm is implemented in distributed system setup and the network-paging latency values of wired and wireless networks are measured experimentally. The experimental results demonstrate that in most cases the algorithm successfully terminates in first round and the possibility of second round execution decreases significantly with the increase in the size of SG ($|N_a|$). The overall message complexity of the algorithm is O($|N_a|$). The comparative study of network-paging latencies indicates that 3G/4G mobile communication systems would support the realization of MDVM system.

Keywords: GPRS, GSM, mobile computing, network paging latency, leader election algorithms

1. Introduction

The availability of portable computing devices having access to WWW, such as smart phones and PDAs, has created a set of high-end mobile applications such as, m-commerce, SES [21] and virtual organization [22]. The mobile devices are limited in hardware resources [24], battery power and operate in doze mode to reduce power requirement [23]. The existing wireless communication technology is restricted in terms of bandwidth and reliability [23]. The existing general-purpose operating systems offer very little support for managing and adapting to the mobile computation paradigm [25]. As a novel approach, the concept of Mobile Distributed Virtual Memory (MDVM) is introduced to enable mobile clients exploiting server resources using the mobile communication network [20]. The Server-Group (SG) of MDVM system allows mobile clients to utilize the server CPU and memory for the data cache and process execution purposes [20]. In this paper, we describe the MDVM system concept, the architecture of mobile communication system consisting of Server-Group (SG) and a randomized distributed algorithm for electing a unique leader and co-leader of the SG in at most two election rounds. The distinguishing features of the proposed algorithm are as followings:

- The proposed distributed randomized algorithm does not assume any statically assigned processor or server IDs. The algorithm assigns the unique IDs to the MDVM servers online.
- The server IDs may not be strictly monotonically increasing or decreasing according to the sequence of registration of the MDVM servers in a SG.
- The algorithm takes at most two rounds to elect a unique leader and co-leader. The experimental results indicate that the algorithm terminates in first round in majority cases and the possibility of entering in the second round of execution decreases significantly with the increase in the size of SG.
- While executing in the second round, the algorithm offers high filter ratio values indicating that the number of MDVM servers participating in the second

round election phase is substantially reduced as compared to the first round of execution. This reduces the network message complexity further.

- The overall message complexity of the algorithm is considerably low and is in the order of the size of SG, $O(|N_a|)$.
- The election rounds are supervised by different coordinators in two different rounds. The coordinator of the second round is elected dynamically. This reduces the possibility of single point of failure.
- Unlike the other leader election algorithms, the proposed algorithm is free from any assumptions such as, static node values, network topology and buffer sizes etc.
- Unlike the probabilistic leader election algorithms proposed earlier, the proposed algorithm does not require several rounds of simultaneous multicasts on the completion of leader and co-leader election in a SG.

We have implemented the algorithm in distributed system setup and have measured the network-paging latency values between a mobile-client and a server in SG. We have compared the paging-latency values for various page sizes in 100Mbps LAN, 10Mbps Wireless-VPN and 2.5G GPRS systems. The experimental results demonstrate that the algorithm terminates in first round in majority cases by electing a unique leader and co-leader however, in few cases the algorithm enters in the second round election phase. The round ratio and filter ratio of the algorithm decrease significantly with the increase in the size of SG. The rest of the paper is organized as followings. Section 2 describes the related work. Section 3 describes the high-end mobile applications along with the technological trends and its limitations. Section 4 illustrates the MDVM system concept and the architecture of mobile communication system containing the SG. Section 5 and section 6 describe the leader election algorithm and the experimental results respectively. Section 7 concludes the paper.

2. Related Works

The leader election algorithms are widely studied in the context of wired and wireless networks. The concept of MDVM system uses the Server-Group (SG) placed in the mobile communication architecture [20]. The SG is consisting of a number of MDVM servers connected by wired TCP/IP network offering resources to the mobile clients. Researchers have proposed a leader election algorithm in a synchronous ring of n servers having predetermined unique identifiers [8, 10]. The algorithm uses $O(n)$ messages if the identifiers are chosen from some countable set. In the worst case, the message complexity of the algorithm is $O(n \log n)$ [10]. It is reported that the deterministic algorithms for the leader election in bidirectional rings [15, 17, 18] and unidirectional rings have $O(n \log n)$ message complexity [10]. These deterministic algorithms work in either synchronous or asynchronous communication models. Although it is shown that the asynchronous communication system leads to $O(n \log n)$ message complexity as a lower bound [17], however, the proof may not be applicable to the synchronous communication model [10]. The one of the difficulties of leader election algorithm proposed in [10] is that it may require a very large number of synchronous rounds in the worst case. In addition, the algorithm assumes statically assigned unique integer IDs to the processors. One of the restrictions of the algorithm is that it discriminates the wake-up-processes from the awakened-processes [10]. Researchers have proposed a self-stabilizing algorithm for leader election in a tree graph [1]. It is illustrated that it may not be possible to design a self-stabilizing algorithm for a tree with the possibility of electing a leaf node as the leader. The proposed algorithm is dependent on the structure of the tree and assumes a predetermined fixed number of nodes in a tree. Another difficulty of the algorithm is that it is dependent on a central daemon and the multiple privileged nodes of the tree should not be adjacent [1]. However, the algorithm does not assume any particular order while choosing a set of active nodes [1]. Another deterministic and self-stabilizing algorithm is proposed in [7] to elect a leader in a ring based on a central daemon. However, the algorithm uses constant space and is self-stabilizing in $O(n^2)$ steps [7]. The algorithm generates a large number of tokens, which increases the complexity of the algorithm considerably. The non-randomized leader election algorithms for the failure-prone asynchronous network can be classified as: *Gallager-Humblet-Spira type algorithms*, *logical network partition based algorithms* and *the algorithms assuming reliable message delivery* [6]. On the other hand, the probabilistic leader election algorithms are generally classified as the randomized algorithms [6, 19]. In general, the message complexity of the probabilistic algorithms varies linearly with the group size and the algorithms require several rounds of $O(N)$ simultaneous multicasts [6]. Hence, this class of probabilistic algorithms may deteriorate the delivery performance of the underlying network. Researchers have proposed a probabilistic leader election algorithm for large groups based on the practical settings [6]. The algorithm offers scalability and assumes week membership in an unreliable network. However, the algorithm is fairly complex and message complexity could be very high depending on the filter value K [6]. In addition, as the view of a process in the group may not be identical for all members of the group, the multiplicity of multicast message rounds may enhance the consumption of network bandwidth considerably, duplicate messaging and may lead to the requirement of message ordering. There exists possibility that N_i value [6] seen by different members may not be consistent and may require multiple circulations. In another approach, a space optimal self-stabilizing leader election protocol for anonymous unidirectional ring is proposed [11]. The algorithm has $O(m_N)$ space complexity [11]. One of the difficulties of the algorithm is its dependence