MAMBO: Membership-Aware Multicast with Bushiness Optimisation

Lu Fan
School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, UK
lf16@hw.ac.uk

Phil Trinder
School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, UK
p.w.trinder@hw.ac.uk

Hamish Taylor
School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, UK
h.taylor@hw.ac.uk

1. INTRODUCTION

Multicast is a key component of applications such as multimedia streaming, multi-player gaming and generic publish-subscribe and event notification services. IP multicast has a number of technological, practical, and business obstacles [3], and thus is not widely used. An alternative, Application-Level Multicast (ALM), has been proposed to support similar functionalities. A survey of existing ALM systems [5] indicates that many ALM systems are built on P2P overlay infrastructures and construct multicast trees among application participants, e.g. Scribe [2] uses Pastry [6].

This article presents Membership-Aware Multicast (MAM) and MAM with Bushiness Optimisation (MAMBO), which reuse the communication structure of a tree-based ALM system to track group membership, i.e. to record when peers join or leave the group. MAM and MAMBO are implemented on Scribe with Pastry and a series of experimental results compare performance with a conventional service-provider architecture. We show that MAM and MAMBO quickly detect, and effectively manage, the arrival and departure of peers. In addition, MAMBO delivers a number of other advantages. The Bushiness Optimisation limits the overhead placed on any peer, hence preserving the scalability of the ALM system. The implementation as Scribe policy plug-ins illustrates that MAMBO can readily be overlapped on any tree-based ALM, and requires few changes to the underlying technology.

2. MOTIVATION

Different P2P applications have varying requirements on membership management, e.g. a file sharing application may not care about the failure of a downloader, but in other applications, a simple "let it fail" strategy may not be appropriate. For example, in a P2P Massively Multiplayer Online Game (MMOG) like Mediator [4], where game participants' computing assets are pooled, the departure of a peer responsible for some computational or administrative tasks must be managed, otherwise it may damage the integrity and consistency of the system.

Similar requirements are also present in P2P incentive mechanisms. P2P applications are by nature voluntary resource sharing systems, in which there is often a tension between individual concerns and collective welfare. In this case, an incentive mechanism tracks the activities of a user in a P2P system, either to bill the user according to the amount of resources that the user has consumed, or to reward the user according to the time and quality of a service that the user has offered to others.

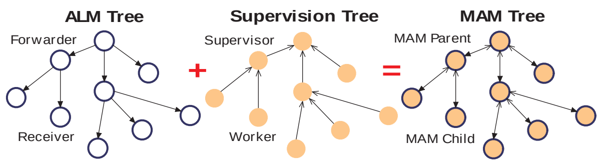

Figure 1: MAM design

Event dissemination and notification are common functionalities that are required by substantial numbers of P2P applications. So, it is helpful to enable an existing ALM system to monitor and organise its members, so as to satisfy the communication and membership management requirements of a P2P application at the same time. Membership-aware multicast can be viewed as a specific kind of reliable multicast, but unlike general reliable multicast [5], the only concern is the arrival and departure of peers.

3. DESIGN

The key idea in MAM and MAMBO is to overlap an ALM *Multicast Tree* with a *Supervision Tree* that monitors and organises its membership, as depicted in Figure 1. Its main novelty is to reuse a communication infrastructure for a secondary purpose, namely membership management.

3.1 MAM Design

The conceptual design of MAM combines ALM and supervision trees. A supervision tree [1] is based on the idea of *Workers* and *Supervisors* - workers do the actual work, and supervisors monitor their behaviour. The design of MAM proposes mapping the logical supervisor-worker relationship to the existing information forwarder-receiver relationship in communication.

3.2 MAMBO Design

Because MAM directly reuses a tree that is created by an ALM system, the shape of a MAM tree may vary. In other words, it is possible for some parents to have many more children than other parents, which results in the former becoming overly busy while the latter are not much used. To address this, a Bushiness Optimisation is proposed to limit the number of children a parent node may accept.

Figure 2 compares the shape of a MAM and a MAMBO tree with at most 3 children. The MAMBO tree is more

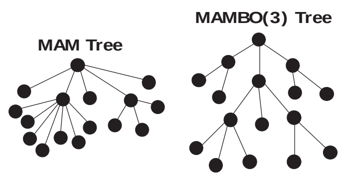

Figure 2: Contrasting MAM & MAMBO trees

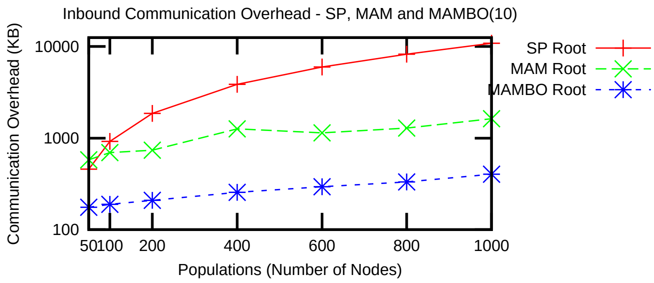

Figure 3: Scalability Comparison

balanced than the MAM tree, and hence the workload is more fairly distributed. A tradeoff is that the MAMBO tree may be deeper and induce greater communication latency, e.g. the same 16-element MAMBO tree has maximum depth 4, where the MAM tree depth is 3.

4. IMPLEMENTATION

MAM, MAMBO and a conventional service-provider (SP) architecture have been implemented using Pastry [6] and Scribe [2]. MAM is implemented as a Maintenance Policy plug-in, which establishes a supervision relationship between each parent and its child nodes. Bushiness Optimisation is implemented as a Scribe Policy plug-in, which entitles an intermediary node to reject adopting a new child when it does not have enough communication bandwidth, or computation power, to forward content to, and monitor the child. Without MAM and MAMBO, a service provider would monitor the working states of all service consumers directly.

5. EXPERIMENTAL RESULTS

A demonstration application, P2P Online Market Place (POMP), is developed to evaluate MAM, MAMBO and a conventional service-provider. The incentive mechanism of POMP bills application participants according to their online time. For various reasons, a participant may fail to log off from the application elegantly. It is a computation and communication intensive task for a service provider to monitor the presence of many participants, and MAMBO is able to distribute such workload among participants.

Figure 3 shows that as the online population increases from 50 to 1000, the overhead of the service provider increases rapidly. However, for the roots of MAM and MAMBO multicast trees, their workloads increase at a much slower speed, which means that MAM and MAMBO are more scalable than a conventional service-provider architecture, and MAMBO is even more scalable than MAM.

Figure 4 displays experimental results for MAM and MAMBO(10) (i.e. nodes have a maximum of 10 children)

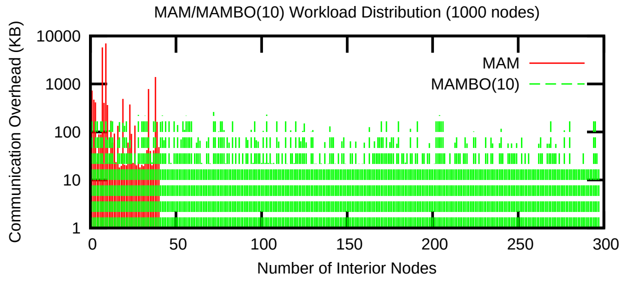

Figure 4: Improved Workload Distribution

workload distribution with 1000 nodes. In MAM, only about 40 nodes were parent nodes, monitoring the working state of the other 1960 nodes, whereas in MAMBO(10), about 300 nodes were parents. It indicates that the shape of the multicast trees built by MAM and MAMBO are quite distinct, and MAMBO more fairly distributes the total workload.

6. CONCLUSION & FUTURE WORK

This article proposes a new membership-aware multicast mechanism, MAMBO, that reuses the ALM infrastructure in a P2P application to monitor and organise its membership by overlapping the multicast tree with a supervision tree.

Both MAM and MAMBO have been implemented as policy plug-ins for Scribe, minimising the changes to the underlying technology while providing membership management for applications like POMP. Experimental results show that MAM and MAMBO are more scalable than a conventional architecture, and MAMBO is more scalable than MAM because of its Bushiness Optimisation.

A limitation of MAM and MAMBO is its inability to detect the failure of a child if its supervising parent fails at the same time. An aim of future work is to provide a secondary policy plug-in that removes this limitation. Furthermore, it is also planned to implement the Mediator multiplayer game framework, and to extend MAMBO with features like hierarchical handling of peer failures.

7. REFERENCES

[1] J. Armstrong. *Making Reliable Distributed Systems in the Presence of Software Errors*. PhD thesis, Swedish Institute of Computer Science, 2003.

[2] M. Castro. SCRIBE: A Large-scale and Decentralized Application-level Multicast Infrastructure. *IEEE Journal on Selected Areas in Communications*, 20(8):1489–1499, 2002.

[3] C. Diot. Deployment Issues for the IP Multicast Service and Architecture. *IEEE Network*, 14:78–88, 2000.

[4] L. Fan, H. Taylor, and P. Trinder. Mediator: A Design Framework for P2P MMOGs. In *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, pages 43–48. ACM, 2007.

[5] M. Hosseini. A Survey of Application-Layer Multicast Protocols. *Communications Surveys and Tutorials, IEEE*, 9(3):58–74, 2007.

[6] A. Rowstron. Pastry: scalable, decentralized object location and routing for large scale p2p systems. In *Proceedings of 18th international conference on distributed systems platforms* ACM, 2001.