Chapter 2.9
A Push-Based Prefetching for Remote Caching RAM Grid

Rui Chu
National Laboratory for Parallel and Distributed Processing, China

Nong Xiao
National Laboratory for Parallel and Distributed Processing, China

Xicheng Lu
National Laboratory for Parallel and Distributed Processing, China

ABSTRACT

As an innovative grid computing technique for sharing the distributed memory resources in a high-speed wide-area network, RAM Grid exploits the distributed computing nodes, and provides remote memory for the user nodes which are short of memory. The performance of RAM Grid is constrained with the expensive network communication cost. In order to hide the latency of remote memory access and improve the performance, the authors proposed the push-based prefetching to enable the memory providers to push the potential useful pages to the user nodes. For each provider, it employs sequential pattern mining techniques, which adapts to the characteristics of memory page access sequences, on locating useful memory pages for prefetching. They have verified the effectiveness of the proposed method through trace-driven simulations.

INTRODUCTION

Grid computing has gained much attention over the past ten years (Foster and Kesselman, 2003; Foster, Kesselman, and Tuecke, 2001). The ultimate goal of grid computing is to share various resources distributed in a wide-area network. Many grid systems have been implemented and deployed (Baru, Moore, Rajasekar, and Wan, 1998; Frey, Tannenbaum, Livny, Foster, and Tuecke, 2001). Most of them shared computing or storage resources successfully based on specific application requirements. RAM Grid (Chu, Xiao, Zhuang, Liu, and Lu, 2006) was our first work that introduced the concept of sharing the vast widely distributed memory resources within the grid. The users of RAM Grid can swap obsolete local memory pages to remote memory instead of

DOI: 10.4018/978-1-4666-0879-5.ch2.9

to the local disk, and the performance of memory intensive applications was boosted due to the lower access time of remote memory than disk when the network transmission is fast enough.

As an extension, we spread the application area of RAM Grid from memory intensive applications to remote caching, which employs widely distributed memory resources as a data cache for local or remote file systems. Our ongoing prototype system named DRACO is designed to be deployed in the distributed, heterogeneous nodes that are connected with a high-speed wide-area network. Using RAM Grid for caching will meet the characteristic of loosely coupled distributed computing environment, which emphasizes to provide "best effort" service, while does not guarantee the degree of performance improvement. In worst case, it is also acceptable that the performance does not raise (and does not drop), while in better network environment, it will gain much more benefits. Nowadays the campus or enterprise network is fast enough to meet the requirements of remote memory sharing, and the rapidly developing network technologies will make our approach more and more attracting.

To facilitate later description, we classify the nodes in RAM Grid (Chu, et al., 2006) into five types. The user node is the consumer of remote memory, while the corresponding memory provider is called the busy node, which comes from the available node. A deputy node serves for one user node, and it acts as a broker and automatically searches available nodes for the user node. The intermediate node does not provide or consume any remote memory. It is ready to become a user node or available node.

In order to study the potential performance improvement, we compare the overheads of data access for an 8KB block over local disk, NFS and RAM grid, which accesses remote memory through the wide area network with 2ms round-trip latency and 2MB bandwidth. From Table 1 we can observe that the caching mechanism in DRACO only reduces the overhead by 25%~30% compared to local disk or NFS access, and the major data access overhead in DRACO mainly comes from the network transmission cost (nearly 60%). Therefore, the performance of DRACO can obviously be more improved if we reduce or hide some of the transmission cost. Prefetching is an approach to hide the cost of low speed media among different levels of storage devices. In this article, we employ prefetching in DRACO in order to improve the performance. Differing from traditional I/O devices, in DRACO, the busy nodes, which provide remote memory for caching, often have extra CPU cycles. Therefore, the busy nodes can decide the prefetching policy and parameters by themselves, thus releasing the user nodes of DRACO, which are often dedicated to mass of computing tasks, from the process of prefetching. In contrast to traditional approaches, in which the prefetching data are decided by a rather simple algorithm in a user node, such a push-based prefetching scheme can be more effective.

RELATED WORK

The basic goal of RAM Grid is to share the plentiful memory resources distributed in a wide area network. As an initial work, several memory sharing schemes, which are usually called network memory systems, have been proposed in 1990s. We can category these systems into three major types based on the objectives, which are high-speed paging device (Feeley, et al., 1995; Markatos and Dramitinos, 1996; Oleszkiewicz, Xiao, and Liu, 2004), data cache for local or networked file systems (Chang and Garcia-Molina, 1999; Dahlin, Wang, Anderson, and Patterson, 1994; Jiang, Petrini, Ding, and Zhang, 2006), or remote RAM disk respectively (Flouris and Markatos, 1999). Unlike network memory schemes, RAM Grid tries to share distributed memory resources in a grid-like environment. It aggregates resources in a large scale and avoids the inadequate idle memory resources problem within a single cluster, while