Dynamic Distributed Load Balancing Support for I/O-Intensive Parallel Applications on Clusters

Faculty: Dr. Hong Jiang and Dr. David Swanson Students: Xiao Qin and Yifeng Zhu

Secure Distributed Information (SDI) Group

Department of Computer Science and Engineering, University of Nebraska-Lincoln

Introduction

- Load balancing schemes can improve system performance by attempting to assign work, at run time, to machines with idle or under-utilized resources.
- Several distributed load-balancing schemes have been presented in the literature, primarily considering CPU, memory, or a combination of CPU and memory,
- The existing schemes have ignored disk I/O.
- The rapidly growing I/O demands of I/O-intensive applications makes storage devices a likely performance bottleneck.
- For any dynamic load balancing scheme to be effective in I/O-intensive application environment, it must be made "I/O-aware".

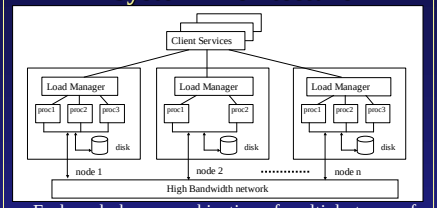
Objective

- Propose a series of simple yet effective I/O-aware load-balancing schemes for both sequential and parallel jobs running on clusters.
- CPU, memory, and I/O resources at each node can be simultaneously well utilized.
- Be adaptive under a wide spectrum of workload.
- Incorporate a preemptive migration technique.
- Feed-back control mechanism is employed.

The Proposed Scheme (IOCM-RE)

- Parallel jobs are assigned to a group of nodes with lighter I/O load. If there are a number of choices, the one with the smallest value of memory load will be chosen to break the tie.
- If no I/O load is imposed on the node, the IOCM-RE scheme considers the node's memory load. When the memory load exceeds the amount of available memory space, the tasks of the newly arrived parallel job are transferred from the overloaded node to the remote nodes that are lightly loaded with respect to memory.
- If both the disk I/O and memory resources of the node are well balanced, the IOCM-RE scheme attempts to evenly distribute the CPU load.

System Architecture



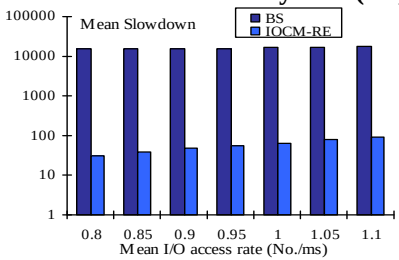
- Each node has a combination of multiple types of resources, such as CPU, memory, network connectivity, and disks.
- Each node maintains a load manager, which is responsible for controlling, monitoring, and load balancing the available resources, in addition to handling the execution of processes generated from the local node.
- Users use client services to interact with the environment

Performance Metric

The performance metric is the mean *slowdown* of all the jobs in a trace. The *slowdown* of a job is defined as the ratio between the job's execution time in a resource-shared setting and its execution time running in the same system but without any resource sharing.

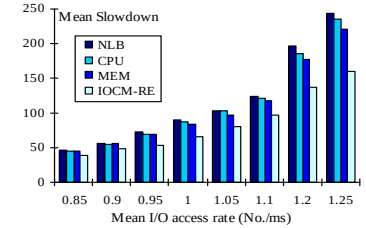
Results

IOCM-RE vs. Batch System (BS)



Mean slowdown as a function of the I/O access rate on the traces with 30% parallel jobs. Page fault rate of 0.5No./ms.

IOCM-RE vs. cpu/memory-based schemes

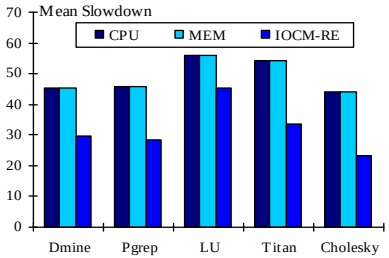


Mean slowdown as a function of the I/O access rate on the traces with 30% parallel jobs. Page fault rate is 0.5 No./ms.

Real I/O-intensive Applications

- Data mining (Dmine): This application extracts association rules from retail data.
- Parallel text search (Pgrep): This application is used for partial match and approximate searches.
- LU decomposition (LU): This application tries to compute the dense LU decomposition of an out-of-core matrix.
- Titan: This is a parallel scientific database for remote-sensing data.
- Sparse Cholesky (Cholesky): This application is capable of computing Cholesky decomposition for sparse, symmetric positive-definite matrices.

Real Applications



Mean slowdowns of four policies on five applications.

Discussion

- IOCM-RE employs remote execution facilities to improve system performance.
- In addition to CPU and memory utilization, IOCM-RE considers I/O load.
- Using five real I/O-intensive applications in addition to a set of synthetic parallel jobs, we have demonstrated that applying IOCM-RE to clusters for I/O-intensive workload is not only necessary but also highly effective.
- When I/O load is low or well balanced, the proposed schemes are able to maintain the same level of performance as the existing non-I/O-aware schemes.
- By combining the proposed approaches with the previous buffer techniques, more intelligent load-balancing decisions can be made.

Future Research

- We have studied the performance of a cluster with 32 nodes. Future research will deal with a rigorous testing experiment where the performance of a cluster with more than 1000 nodes will be evaluated.
- Since conducting this experiment largely depends on the simulator, we will make an effort to develop a parallel simulator to efficiently investigate the performance of the I/O-aware load balancing schemes for large-scale clusters.

References

1. X. Qin, H. Jiang, Y. Zhu, and D. Swanson, "Towards Load Balancing Support for I/O-Intensive Parallel Jobs in a Cluster," *Proc. of IEEE Int'l Conf. on Cluster Computing 2003.*
2. X. Qin, H. Jiang, Y. Zhu, and D. Swanson, "Dynamic Load Balancing for I/O-Intensive Tasks on Heterogeneous Clusters," *Proc. of the 10th Int'l Conf. on High Performance Computing* (HiPC 2003), Dec.17-20, 2003.
3. Y. Zhu, H. Jiang, X. Qin, and D. Swanson, "A Case Study of Parallel I/O for Biological Sequence Analysis on Linux Clusters", *Proc. of IEEE Int'l Conf. on Cluster Computing 2003.*
4. X. Qin, H. Jiang, Y. Zhu, and D. Swanson, "Dynamic Load balancing for I/O- and Memory-Intensive workload in Clusters using a Feedback Control Mechanism," *Proc. of the 9th Int'l Euro-Par Conf. on Parallel Processing (Euro-Par 2003)*, 2003.
5. M. Harchol-Balter and A. Downey, "Exploiting Process Lifetime Distributions for Load Balacing," *ACM transaction on Computer Systems*, vol. 3, no. 31, 1997.
6. X. Zhang, Y. Qu, and L. Xiao, "Improving Distributed Wrokload Performance by Sharing both CPU and Memory Resources," *Proc. 20th Int'l Conf. Distributed Computing Systems (ICDCS 2000)*, Apr. 2000.