On Improving Performance and Conserving Power in Cluster-based Web Servers

Heung Ki Lee, Gopinath Vageesan and Eun Jung Kim
Texas A&M University
heungkilee@neo.tamu.edu,vgopi@ee.tamu.edu,ejkim@cs.tamu.edu

Abstract

With the growing use of cluster systems in web servers, file distribution and database transactions, power conservation and efficiency have been identified as critical issues in the design of cluster systems. Widely adopted, distributor-based systems forward client requests to a balanced set of backend servers in complete transparency to the clients. In this paper, we use power and locality-based request distribution at the distributor to provide optimum power conservation, while maintaining the required QoS of the system. The distribution scheme uses a simple memory management technique using pinned memory on the backend servers and proactive distribution, with the aid of data organization of the website, to improve the locality of the files. A simple on-off based power management scheme is applied to conserve power. Our scheme provides reduced response time to the clients and improved power conservation at the backend server cluster without compromising performance. Simulations involving real-time web traces and latest web technologies witness performance boost of 15-23% and power conservation of 15-48% over the existing policies.

Power and Locality-aware Request Distribution (PLARD)

Cluster systems are being increasingly used in the web server management, file distribution and database transactions. The main reason for the large-scale deployment of the cluster systems is their load sharing and high performance capabilities. However, it has been observed that larger scale cluster-based servers contribute to approximately 40% of the overall delay, and this delay is likely to grow with the increasing use of cluster nodes and dynamic content. In addition, cluster servers consume significant power, and the power usage is a major fraction of the total ownership cost.

Among the different architectures in the cluster-based web servers, the distributor-based systems have been widely deployed. The requests are forwarded to the set of backend servers based on a certain policy. LARD (Locality-Aware Request Distribution), PARD (Power-Aware Request Distribution) and WRR (Weighted Round Robin) are few of the most prolifically adopted policies. They focus on improving efficiency, power conservation and load balancing respectively. It has been proven that locality-based request distribution schemes bring exceedingly good performance boost to distributor-based cluster systems. But, such systems do not have any scheme for power conservation. Similarly, Power-aware request distribution (PARD) scheme focuses on conserving the power by turning the servers OFF and ON depending on the load on the servers, but suffer from a performance viewpoint.

To overcome this shortcoming in the design, we propose a Power and Locality-aware request distribution scheme (PLARD). But, as there is a tradeoff between power and performance factors, our goal is to improve the performance with a state of considerable power conservation. Thus, we focus on improving the locality of the files that are being requested. By increasing the locality, we can achieve better hit rates in the backend servers' memory and hence a performance boost.

To achieve this, we use two techniques: the application-level memory management technique using pinned memory and proactive distribution with the aid of the data organization of the website. We use pinned memory to preserve the locality of the files when the servers are turned OFF by the power management policy. The data in the pinned memory can be migrated across the working backend servers on the event of turning OFF a server. This preserves the locality of the files.

Additionally, we analyze the organization of the website and the website's log files to group file requests that are always bound to be requested by the

Proceedings of the IEEE International Conference on Web Services (ICWS'05)
0-7695-2409-5/05 $20.00 **IEEE**

client together. For example, the webpage and its embedded objects are designated a group and we employ a simple pre-fetching scheme to pre-fetch the embedded objects of a requested page into the pinned memory, even before the request for the object arrives at the backend. These two schemes improve the service time of the files and hence boost the overall performance of the system. Figures 1 and 2 illustrate the working of the scheme.

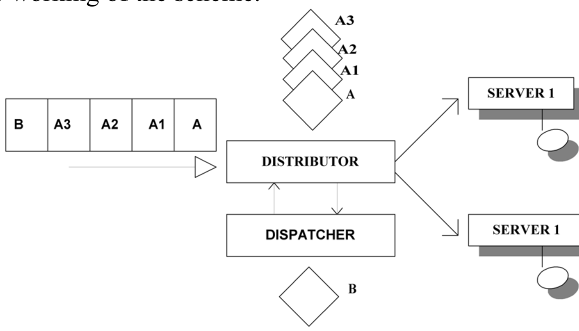


Figure 1: PLARD Policy: The requests are distributed such that the requested page and its embedded objects are served by the same server on the persistent HTTP connection.

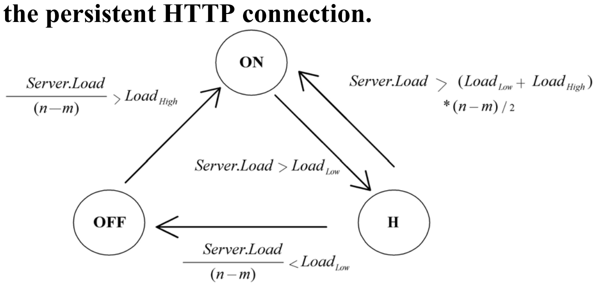


Figure 2: Power Transition States of the Server: The load-based policy to turn OFF, turn ON and hibernate servers.

The following policies are used as the benchmark to compare the final results: WRR, LARD, PARD and techniques proposed earlier for HTTP 1.1 (Ext-LARD-PHTTP). The simulations are carried out using real-time web traces of various websites.

Figure 3 shows the efficiency comparison of the dispatchers of all the schemes. As PLARD-PHTTP reduces the overhead at the dispatcher by proactively switching the related requests (requested file and embedded objects) to a single server, the number of dispatches required is very less compared to LARD policies.

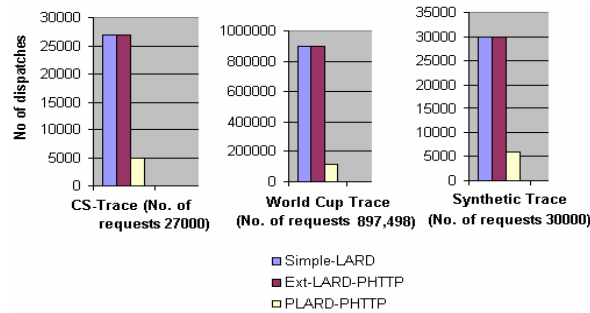
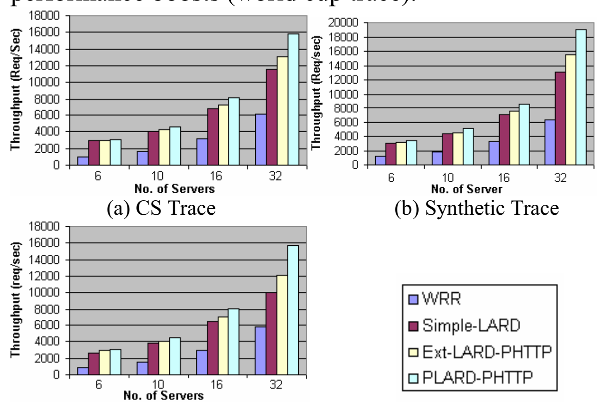


Figure 3: Dispatcher Efficiency

Figure 4 shows the Throughput of the various policies, with the 3 different web traces. It can be noted that PLARD-PHTTP achieves a 15% (CS trace) and 23% performance boosts (world cup trace).



(a) CS Trace (b) Synthetic Trace

(c) World Cup Trace

Figure 4: Throughput Comparison

Both these improvements can be attributed to the minimum overhead at the dispatcher.

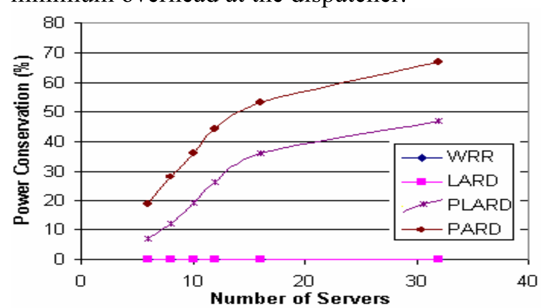


Figure 5: Power Conservation

Figure 5 shows our policy provides power conservation up to 48%, which will halve the operating power budget of large cluster-based servers. With increase in the number of servers, the PLARD system tries to optimize the power consumption by turning OFF/hibernate more servers.

In summary, our PLARD scheme provides performance boost of 15-23% and power conservation of 15-45% over the existing policies.