Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Task assignment policies in distributed server systems: A survey

Fouzi Semchedine [*], Louiza Bouallouche-Medjkoune, Djamil Aïssani

LAMOS, LAboratory of Modeling and Optimization of Systems and Doctoral School in Computer Science, (Networking and Distributed Systems), University of Béjaïa, 06000, Algeria

ARTICLE INFO

Article history:
Received 13 September 2010
Received in revised form
7 December 2010
Accepted 21 January 2011
Available online 31 January 2011

Keywords:
Data intensive computing
Task assignment
Distributed server systems
Heavy-tailed distribution
Load balancing
Load sharing
Performance

ABSTRACT

Data intensive computing, in the Web environment, motivates the distributed designs of Web server systems (Web clusters) because of their scalability and cost-effectiveness instead of one Web server with high performance. The task assignment policy, in such systems, focuses on the manner of assigning the tasks that reach these systems (e.g. the case of intensive requests that reach the distributed Web server systems) in order to minimize the response time and thus, improve the performance. These tasks, generally, follow the "heavy-tailed" distribution which has the property that there is a tiny fraction (about 3%) of large tasks that makes half (50%) of the total load. Several policies were proposed in the literature to deal with the nature of this Web traffic. This paper presents a state-of-art of the existing task assignment policies. We classify these policies in two classes: policies which assume that the task size is known a priori, and policies which assume that the task size is not known a priori (like TAGS, TAPTF and TAPTF-WC). The first class of policies regroups policies which consider no knowledge of load information at the servers when assigning the incoming tasks, known as static policies (like Random, Round Robin, etc.) and, policies, known as dynamic policies (like Central Queue Policy, Least Loaded First "LLF", etc.) which use some load information (e.g. the processing capacity, the queue load, etc.) to process.

Contents

1. Introduction

The explosion of the Internet and the World Wide Web has sensibly increased the amount of the online information and services available for the Web users. The growing of information and service demands has placed a dramatic pressure on the Internet infrastructure by the need of advanced Web server systems capable of serving a large number of Web requests. Hence, the distributed designs of Web server systems (Web clusters) appear because of their scalability and cost-effectiveness, instead of one Web server with high performance like a mainframe.

Figure 1 illustrates a typical architecture of a distributed server model. In this architecture, one virtual IP address (VIP) is assigned to the Web clusters, which is the IP address of the dispatcher. This is able to identify each server in the cluster through a private address and distributes the load among the Web servers based on the mechanism used to route the requests. Furthermore, the selected Web server sends the response packets to the client.

The tasks (we refer to tasks as requests for Web files) arrive to the distributed server system following the Poisson process with rate λ (Nozaki and Ress, 1978; Williams et al., 2005) and must be

[*] Corresponding author at: LAMOS Laboratory, University of Béjaïa, 06000, Algeria. Tel.: +213 550 493 611; fax: +213 34 21 51 88.
E-mail addresses: fouzi.jams@gmail.com, fouzi_jams@yahoo.fr (F. Semchedine).

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.01.011