Chapter 11
Web Service Clustering and Data Mining in SOA System

Sreeparna Mukherjee
National Institute of Technology Surathkal, India

Asoke Nath
St. Xavier’s College (Autonomous), India

ABSTRACT

The success of the web depended on the fact that it was simple and ubiquitous. Over the years, the web has evolved to become not only the repository for accessing information but also for storing software components. This transformation resulted in increased business needs and with the availability of huge volumes of data and the continuous evolution in Web services functions derive the need of application of data mining in the Web service domain. Here we focus on applying various data mining techniques to the cluster web services to improve the Web service discovery process. We end this with the various challenges that are faced in this process of data mining of web services.

INTRODUCTION

In the beginning of web, the accomplishment of the web relied on upon the way that it was straightforward and universal as it just conveyed static HTML based website pages. However, as static pages neglected to take into account the dynamic client demands, it was immediately obsolete, and content management of websites became very vital. Throughout the years, the web has advanced to end up the storehouse for getting to data as well as for storing software.

The Common Gateway Interface (CGI) was introduced for providing as two-tier web applications for providing dynamic content to the users. The CGI acts as client by retrieving content from external resources, such as a database. Here CGI as a client in a traditional system.

However, CGI suffered from various major drawbacks, such as:

1. Since the database was running on the same machine, it became difficult to make back-ups.
2. It suffered from context-switching problems as it was running as a separate process.

DOI: 10.4018/978-1-5225-2157-0.ch011
3. It had design flaws which affected performance, scalability as well as security. (Kiet T. Tran, PhD (2013))

Hence, CGI implemented the traditional Centralized model where all computing was done in a single machine and all the computing resources in the primary datacenter. This includes Domain Authentication Servers, Share Files, Emails as well as Applications. Although centralized model had the benefit of lower operational cost as well as very little complexity, it has a lot of disadvantages where the remote server’s WAN became the most frequent single point of failure.

In a purely distributed model, every site is self-maintained generally. While some availability to the essential datacenter is required, the remote website would have its own particular Email Server, deal with its own particular reinforcements, control its own Internet access, and have its own particular Shared Files. Application access may, in any case, depend on HQ, albeit numerous applications support this kind of appropriated model.

The advantage of a Distributed model is that every site can “get by” all alone. There is no Single Point of Failure in such manner. Likewise, accepting that the equipment in a portion of the locales are put away in a protected Server Room and not with the workplace supplies, this additionally would possibly encourage Business Continuity by using Sites that reference each other as possible sites.

The drawback to this methodology, clearly, is expense. In addition to the fact that this would require extra equipment and programming costs, yet one unquestionably would require no less than an incomplete on location nearness at every area paying little mind to what number of remote administration parts are set up. Another thought would be the reinforcement engineering. Unless every site had a solid measure of transmission capacity, in any event, the underlying information reinforcement preparing would need to be taken care of locally before being sent or imitated offsite. (Eric Dosal (2005))

A three-tier architecture is a client server architecture which has a user interaction layer, the business rules layer, and data services (including databases) layer.

The three tiers in a three-tier architecture are:

- **Presentation Tier:** This tier is on the top level of the architecture. The main functionality is to display information related to services available and also to communicate with the rest of tiers. The communication is done by sending results in between the browser as well as the other tiers.
- **Application Tier:** Popularly known as the middle tier, the main function of this tier to control application functions by doing an elaborate processing.
- **Data Tier:** As the name suggests, this tier is responsible for storing information and efficient retrieval of stored data. (Microsoft)