Chapter 15
The Need to Consider Hardware Selection when Designing Big Data Applications Supported by Metadata

Nathan Regola
University of Notre Dame, USA

David A. Cieslak
Aunalytics, Inc., USA

Nitesh V. Chawla
University of Notre Dame, USA

ABSTRACT

The selection of hardware to support big data systems is complex. Even defining the term "big data" is difficult. "Big data" can mean a large volume of data in a database, a MapReduce cluster that processes data, analytics and reporting applications that must access large datasets to operate, algorithms that can effectively operate on large datasets, or even basic scripts that produce a needed resulted by leveraging data. Big data systems can be composed of many component systems. For these reasons, it appears difficult to create a universal, representative benchmark that approximates a "big data" workload. Along with the trend to utilize large datasets and sophisticated tools to analyze data, the trend of cloud computing has emerged as an effective method of leasing compute time. This chapter explores some of the issues at the intersection of virtualized computing (since cloud computing often uses virtual machines), metadata stores, and big data. Metadata is important because it enables many applications and users to access datasets and effectively use them without relying on extensive knowledge from humans about the data.

DOI: 10.4018/978-1-4666-4699-5.ch015

INTRODUCTION

Big data systems have emerged as a set of hardware and software solutions to allow organizations to obtain value from the increasing volume and complexity of data that is captured. Web sites are one example of leveraging big data systems and data to improve key business metrics. For example, large organizations often have a portal site that is a vital part of their business operations, whether it a private extranet, a public site for news, or an e-commerce site. While major public portal sites often appear that they are one seamless site, they are often built from many separate applications. For example, a news site may have an application that lists the top ten most popular news articles or an ecommerce site may recommend products for a user. Increasingly, these applications or components are driven by big data systems. Major portal sites are essentially large distributed systems that appear as a seamless site. This approach allows the operators of the site to experiment with new applications and deploy new applications without impacting existing functionality. It also importantly enables the workflow of each application to be distinct and supported by the necessary hardware and application level redundancy that is appropriate for that specific application. An e-commerce site would likely invest substantial resources to ensure that the "shopping cart" application was available with an uptime of 100%, while the "recommended products" application may have a target uptime of 99.9%. Likewise, a news site may decide that the front page should be accessible with a target uptime of 100%, even during periods of extreme load such as on a national election day. The news site may decide that the "most popular articles" application should have a target uptime of 99.99%. For example, a small pool of servers may present the content for display that is produced by each application, but each application may have its own internal database servers, Hadoop cluster, or caching layer.

News Site Use Case

Assume that a news site began their "most popular articles application" by ranking the frequency of news article displays. The intent of the "most popular articles application" on a news site is often to increase the length of a site visit and this is naturally linked to the business objectives of an organization. If the news site is revenue driven then increasing the length of a site visit increases the likelihood that the user will click on advertisements that generate revenue for the owner. Given this background, assume that in the initial version of the "most popular articles" application, users in the southern United States caused the most popular article to be an article that discussed an impending hurricane that might hit the southern United States. Users in the central United States may only rarely read this "most popular article" since it is not relevant to users in the central United States. Calculating the most popular articles on a news site is relatively easy, assuming that a well designed database of activity exists. However, this simple calculation isn't necessarily optimal from a business metric standpoint. The user should be presented the "most popular articles" that are actually relevant or are likely to be clicked on, since the goal is to increase traffic, not just present a ranking of the most read articles to the user. Perhaps the data scientist discovers that weather articles are generally sought out by users that are aware of impending weather events or are rarely read outside of the immediate geographic area of the weather event. The data scientist may decide to utilize information concerning the type of the article and the geographic area discussed by an article to determine whether it should be present in a "most popular articles list." Both enhancements require more complex data and a more complex query. For example, in the simplest revision of the "most popular articles" application, articles must be tagged with the type of article: "weather" or "non-weather" and the geographic area discussed