Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 50 (2015) 517 – 523

Procedia Computer Science

2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Survey on Programming Models and Environments for Cluster, Cloud, and Grid Computing that defends Big Data

J. Christy Jackson[a], V. Vijayakumar[b], Md. Abdul Quadir[c], C. Bharathi[d]

[a]*J.Christy Jackson, VIT UNIVERISITY-CHENNAI 600063,INDIA*
[b]*V. Vijayakumar, VIT UNIVERISITY-CHENNAI 600063,INDIA*
[c]*Md.Abdul Quadir, VIT UNIVERISITY-CHENNAI 600063,INDIA*
[d]*C.Bharathi, VIT UNIVERISITY-CHENNAI 600063,INDIA*

Abstract

A collection of interlinked stand-alone computers which function together in unity as a single incorporated computing resource is a kind of parallel or distributed processing systems called as the cluster. Clusters and grids are systems which intercommunicate among them and act as a single resource. This type of functionality can be referred to as the multi-computer parallel architecture that runs on certain considerations. Nevertheless cloud computing came forth as a more illustrious programming model to handle large data sets using clusters. A programming model is nothing but how data is carried out for handling the application. Performances, portability, objective architecture, sustainment of code are key measures that have to be commemorated while designing a programming model. Applications which are meant for data analytics generally deal with large data sets that undergo several stages of processing. Some of these stages are performed consecutively, and the others are carried out in parallel on clusters, grids, and cloud. This paper portrays a survey on how programming models which are developed for cluster cloud and grid act as a support for big data analytics. In addition, study on programming models which are currently being employed by leading multinational companies are pictured in the paper.

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Keywords:

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)
doi:10.1016/j.procs.2015.04.025

1. Introduction

Breakthroughs in industrial innovations, next generation scientific discoveries will depend on the capability of systems to analyse on the large volumes of data available. As these masses of information grow rapidly, the challenge faced would be on how to manage these large chunks of data, which in turn causes an increase in complexity of Data Life Cycle management (DLM). DLM comprises of various operations such as transferring, archiving, replication, processing, and deletion. Results are required to automate and enhance data management operations in order to ease the complexity of Data Life Cycle. It is observed that Data Life Cycle is influenced by two constraints. The first constraint being the operations on data which are extracted from the users and applications and the second constraint is the infrastructure itself. The second challenge banks on the fact that data are distributed across variety of systems and infrastructure and not just one single infrastructure [9]. Hence Big Data Applications have to be able to coordinate several systems which address the data and also has to deal with consequences related to data and the events happening. This paper portrays around the second constraint, which is infrastructure itself and is scripted with a complete analyses on programming models and environments which support Big Data infrastructure.

2. Programming Models

Programming model can be determined as the stream and performance of the data manipulation for an application. Execution, portability, target architectures, ease of sustenance code revision mechanisms are some of the things to be considered while developing a programming model. More often than not some of these factors have to be compromised for service. A typical example would be trading computation for storage or for communication of data is an extremely common algorithmic manipulation. These complications can be overcome by using parallel algorithms and hardware. Surely, an application developer or the programmer may have multiple cases of similar algorithm to allow for various performance tuning on different varieties of hardware architecture [4].

Hardware architectures these days are small and high-end high performance computer which form clusters with versatile communications, interconnection technologies and with nodes having processor greater than one. An Illustration for this architecture can be the Earth simulator which is a cluster of enormously powerful nodes with multiple vector processors and with prominent IBM space installation. These simulators have multiple nodes with 4, 8, 16, 32 processors each [4]. The issue arises when a situation originates the requirement of selecting a programming model that acquires the data in the correct place when computational resources are useable. As the number of processors grows, this problem becomes more complex. Scalability is the term employed to suggest the performance of an algorithm, method, or code, relative to single processor. The scalability of an application is principally the consequence of the algorithms capsuled in the programming model which is used in the application [4].

2.1 Active Data Programming

The life cycle of data is the track of operational stages through which data passes from the time they enter the system until they leave the system. Amongst these two points in time, data actually passes through various stages of development. Migration, archiving, duplication, transfer are some of the stages through which data mostly passes. Active Data Life Cycle management is a programming model which assists the developers in writing of applications which implement data life cycle management [9].

2.1.1. Prototyping Life Cycles

Active Data is a mode to model life cycles. This life cycle model is primarily based on Petri networks. Petri Networks are a formal graphical tool extensively used for the analysis of systems with concurrency and resource sharing. Each data item in the data set is tagged with a state. With respect to these states all possible data states with places are represented. As it is common for distributed systems to deal with data reproduction, a petri net