

Paper 1

Vouk, M.A. "Cloud Computing – Issues, Research and Implementations" 30th International Conference on Information Technology Interfaces (2008): 31-40.

Summary

The research paper “Cloud Computing – Issues, Research and Implementations” defines the concepts and hierarchy associated with cloud computing. It also presents North Carolina State University's experiences with their own implementation of an opensource cloud computing project they call the Virtual Computing Lab (VCL).

The paper defines four major concepts of cloud computing. These are: Service-oriented Architecture, Service Components, Workflows and Virtualization. SOA provides the framework for communications. Service components define a building-block approach that can enable reusability, substitution, extensibility and more. Workflows define the directed graphs necessary for coordinating work between components. And finally, virtualization, of which there are many solutions, enables one machine to serve many purposes.

The paper lays out a hierarchy of development and feedback that flows from users to provisioning to service authors to “cyberinfrastructure” developers and back. Each of these hierarchical levels can have explicit domain expertise without having to have deep knowledge of the other levels.

Finally, the paper documents North Carolina State University's implementation of VCL. Currently, VCL serves a student and faculty population of 30,000 with facilities for both generic server images and specialized images. These specialized images can be used to implement an opensource version of MapReduce or a subcloud for grid computing.

VCL is helping researchers at NC State to investigate issues such as image and service construction, image meta-data needs, cloud provenance especially the management and coordination of resources, optimization and image portability. Usage of their system has grown steadily and the real world nature of their implementation has sped feedback on their research interests.

Paper 2

Buyya, Rajkumar, Chee Shin Yeo and Srikumar Venugopal "Market Oriented Cloud Computing"
Proceedings of the 2008 10th IEEE International Conference (2008): 5-13.

Summary

The research paper “Market Oriented Cloud Computing” foresees connected cloud computing resources and the development of a market-oriented approach for matching computing needs with the best provider based on price, performance and quality guarantees. It stems from the original concept of “computer utilities” which work like present day electrical or phone utilities except that the deliverable is computing resources.

In this paper, the authors see a market-driven hierarchy with users and brokers interacting with allocators that reconcile availability, service level agreement needs, pricing and monitoring. These allocators then invoke the necessary virtual machines and physical machines necessary to provide the needed response. Today this hierarchy only exists at the lower levels since there are no standards among current cloud providers.

The paper sees the allocation process as being the most critical since it must address a wide variety of issues to provide the best allocation of computing need to cloud resources. These issues include: service management, risk management, resource management, pricing, accounting, monitoring and dispatching.

This paper summarizes the cloud and grid offerings from several providers and helps identify the many differences between the offerings both from a technical standpoint and a consumer standpoint. The authors see the need for standards and interconnections to enable a true market-driven approach. They also see the rise of brokers who help match consumers and providers based on need including the use of brokers as a channel for cloud delivery from multiple providers.

Paper 3

Nurmi, Daniel, Rich Wolski, Chris Grzegorzcyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. "The Eucalyptus Open-Source Cloud-Computing System". Proceedings of Cloud Computing and Its Applications (2008).

Summary

The research paper "The Eucalyptus Open-Source Cloud Computing System" presents Eucalyptus an cloud computing opensource software framework developed at the University of California, Santa Barbara. The paper outlines the software's design and the tradeoffs that were made to make it more appropriate to an academic setting.

Based on an emulation of Amazon's EC2 interfaces, Eucalyptus allows users to start, control and stop virtual machine instances. The software is designed around three primary components including an *Instance Manager* that controls virtual machine instances, a *Group Manager* that manages, reports and schedules instance managers and a *Cloud Manager* that serves as the main interface for users and administrators.

Each of the three main components of Eucalyptus have their own API's surfaced through a WSDL document. One of the unique elements of Eucalyptus is its use of software from the Virtual Distributed Ethernet project. This project implements the Ethernet protocol in software so that virtual switches and cables and be run to specific instances. This virtual Ethernet environment provides the ability to completely "rewire" instances and managers as needed.

Eucalyptus is designed to run in a heterogeneous environment and as such provides additional support for surveying, allocating and deallocating varying resources that might be typical in an academic setting. The software implements zones of need that are mapped to clusters of virtual machines.

It is intended that Eucalyptus provide an experimentation platform that is easy to deploy on a variety of hardware platforms. The system is currently being used to research high performance computing and the potential of combining cloud systems.

Paper 4

Dean, Jeffrey, Sanjay Ghemawat ”[MapReduce: Simplified Data Processing on Large Clusters.](#)”
Communications of the ACM (2008): 107-113.

Summary

The research paper “MapReduce: Simplified Data Processing on Large Clusters” describes the programming framework that Google developed to parallelize work across a large number of machines. Their approach called “MapReduce” deals with issues such as work distribution, fault handling and result summarization in such a way that lets developers solve problems by simply defining a “map” function and a “reduce” function.

The “map” function takes a unit a work, computes results and emits key-value pairs of answers. The “reduce” function takes the key-value pairs generated by the map function and summarizes the answers into a smaller set of key-value pairs.

While the basic map and reduce and functions can be quite small, Google's implementation of MapReduce as a framework has become increasingly sophisticated as new challenges and needs have arisen. These challenges included: efficient distribution of mappers and reducers, minimizing network traffic between nodes, debugging, fault tolerance including failures by particular worker nodes and data ordering guarantees. By creating a framework to execute MapReduce functions, the creation of programs that are able to work against a broad collection of data has been greatly simplified.

The paper concludes with Google's own experiences with MapReduce. In particular, it discusses how MapReduce has been used to redesign its document indexing enabling faster processing and easier changes to algorithms. In general, the paper mentions how MapReduce has been used for large-scale machine learning problems, clustering problems, creating report summaries, processing satellite imagery, language model processing for machine translation and more.

Paper 5

Chang, Fay, Jeffrey Dean, Sanjay Ghemawat and Wilson Hsieh "Bigtable: A Distributed Storage System For Structured Data" ACM Transactions on Computer Systems Volume 26 Issue 2 (June 2008): Article #4.

Summary

The research paper "Bigtable: A Distributed Storage System for Structured Data" describes the distributed storage system that Google has designed to handle the petabytes of data they have collected in their business. Although some of the terms and approach may resemble a typical RDMS, Bigtable serves to address a more specialized data problem where a large amount of textual data is spread out across thousands of machines.

Bigtable treats all data as strings and lets clients dynamically control data layout and format and provides them control over the locality of data based on their implementation of a particular schema. A data table in Bigtable is three dimensional representation that is determined by a row identifier, column identifier and a timestamp. The row and column identifiers are strings that can be up to 64KB in size. A cell identified by a row and column identifier is timestamped so that new versions of data do not override previous versions. Bigtable provides the ability to limit the number of versions kept and provides background garbage collection to remove unneeded versions of data.

Rows are grouped into tablets with similar row keys in the same tablet. These tablets are then managed in a B-Tree style organization by a separate tablet file system called Chubby that works with multiple replicas of the data to provide data fault tolerance. Groups of tablets may be distributed across machines to provide a distribution of work so that one machine is not overly critical to data retrieval.

Bigtable has proven very successful for Google. It has provided them high performance and high availability across over sixty Google products and services. Because of the large amount of data and the large number of machines involved, fault tolerance has become a major feature of the software. Failures that had to be dealt with included: memory and network corruption, large clock differences, hung machines, quota overflows and planned and unplanned hardware maintenance.