Cloud Based Dynamic Workload Management

Ms. Betsy M Babykutty[1], Mr. K. Chandramohan [2]

PG Student
Affiliated to Anna University Chennai, Dept. of Computer Science and Engineering,
Gnanamani College of Engineering
betzybaby009@gmail.com

[2]Head of the Dept. of Computer Science and Engineering,
Gnanamani College of Engineering
info@gce.org.in

Abstract: Dynamic resources allocation system, that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. But, this allocation storage had a limits to store a data. The allocation process is very complexity and it needs more server to allocate the resources, so it's very costly to maintain. To overcome this problem, we proposed, a data compression with the use of j-bit encoding (JBE). This algorithm will manipulates each bit of data inside file to minimize the size without losing any data after decoding which is classified to lossless compression.

Keywords: about four key words separated by commas**.**

1. Introduction

The elasticity and the lack of upfront capital investment offered by cloud computing is appealing to many businesses. There is a lot of discussion on the benefits and costs of the cloud model and on how to move legacy applications onto the cloud platform. Here we study a different problem: how can a cloud service provider best multiplex its virtual resources onto the physical hardware? This is important because much of the touted gains in the cloud model come from such multiplexing. Studies have found that servers in many existing data centers are often severely underutilized due to overprovisioning for the peak demand. The cloud model is expected to make such practice unnecessary by offering automatic scale up and down in response to load variation. Besides reducing the hardware cost, it also saves on electricity which contributes to a significant portion of the operational expenses in large data centers.

Proposed system

In this paper, we explore the problem of sharing a cluster between users while preserving the efficiency of systems like Map Reduce – specifically, preserving data locality, the placement of computation near its input data. Locality is crucial for performance in large clusters because network bisection bandwidth becomes a bottleneck. Our work was originally motivated by the Map Reduce workload at Facebook. Event logs from Face book's website are imported into a 600-node Hadoop data warehouse, where they are used for a variety of applications, including business intelligence, spam detection, and ad optimization.

Existing System

Desktop computers are an attractive focus for energy savings as they are both a substantial component of enterprise energy consumption and are frequently unused or otherwise idle. Indeed, past studies have shown large power savings if such machines could simply be powered down when not in use. Unfortunately, while contemporary hardware supports low power "sleep" modes of operation, their use in desktop PCs has been curtailed by application expectations of "always on" network connectivity.

LITERATURE REVIEW

1.Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling

As organizations start to use data-intensive cluster computing systems like Hadoop and Dryad for more applications, there is a growing need to share clusters between users. However, there is a conflict between fairness in scheduling and data locality (placing tasks on nodes that contain their input data). We illustrate this problem through our experience designing a fair scheduler for a 600-node Hadoop cluster at Facebook. To address the conflict between locality and fairness, we propose a simple algorithm called delay scheduling: when the job that should be scheduled next according to fairness cannot launch a local task, it waits for a small amount of time, letting other jobs launch tasks instead. We and that delay scheduling achieves nearly optimal data locality in a variety of workloads and can increase throughput by up to 2x while preserving fairness. In addition, the simplicity of delay scheduling makes it applicable under a wide variety of scheduling policies beyond fair sharing.

2.SleepServer: A Software-Only Approach for Reducing the Energy Consumption of PCs within Enterprise Environments.

Desktop computers are an attractive focus for energy savings as they are both a substantial component of enterprise energy consumption and are frequently unused or otherwise idle. Indeed, past studies have shown large power savings if such machines could simply be powered down when not in use. Unfortunately, while contemporary hardware supports low power "sleep" modes of operation, their use in desktop PCs has been curtailed by application expectations of "always on" network connectivity. In this paper, we describe the architecture and implementation of SleepServer, a system that enables hosts to transition to such low-power sleep states while still maintaining their application's expected network presence using an on demand proxy server. Our approach is particularly informed by our focus on practical deployment and thus SleepServer is designed to be compatible with existing networking infrastructure, host hardware and operating systems. Using SleepServer does not require any hardware additions to the end hosts themselves, and can be supported purely by additional software running on the systems under management. We detail results from our experience in deploying SleepServer in a medium scale enterprise with a sample set of thirty machines instrumented to provide accurate real-time measurements of energy consumption. Our measurements show significant energy savings for PCs ranging from 60%-80%, depending on their use model.

3.Quincy: Fair Scheduling for Distributed Computing Clusters

This paper addresses the problem of scheduling concurrent jobs on clusters where application data is stored on the computing nodes. This setting, in which scheduling computations close to their data is crucial for performance, is increasingly common and arises in systems such as Map Reduce, Hadoop, and Dryad as well as many grid-computing environments. We argue that data intensive computation beets from a one-grain resource sharing model that differs from the coarser semi-static resource allocations implemented by most existing cluster computing architectures. The problem of scheduling with locality and fairness constraints has not previously been extensively studied under this model of resource sharing. We introduce a powerful and exile new framework for scheduling concurrent distributed jobs with one-grain resource sharing. The scheduling problem is mapped to a graph data structure, where edge weights and capacities encode the competing demands of data locality, fairness, and starvation-freedom, and a standard solver computes the optimal online schedule according to a global cost model. We evaluate our implementation of this framework, which we call Quincy, on a cluster of a few hundred computers using a varied workload of data- and CPU-intensive jobs. We evaluate Quincy against an existing queue-based algorithm and implement several policies for each scheduler, with and without fairness constraints. Quincy gets better fairness when fairness is requested, while substantially improving data locality. The volume of data transferred across the cluster is reduced by up to a factor of 3.9 in our experiments, leading to a throughput increase of up to 40%.

4.Map reduce Optimization Using Regulated Dynamic Prioritization

We present a system for allocating resources in shared data and compute clusters that improves Map Reduce job scheduling in three ways. First, the system uses regulated and user-assigned priorities to offer different service levels to jobs and users over time. Second, the system dynamically adjusts resource allocations to at the requirements of different job stages. Finally, the system automatically detects and eliminates bottlenecks within a job. We show experimentally using real applications that users can optimize not only job execution time but also the cost-Benet ratio or prioritization efficiency of a job using these three strategies. Our approach relies on a proportional share mechanism that continuously allocates virtual machine resources. Our experimental results show a 11−31% improvement in completion time and 4−187% improvement in prioritization efficiency for different classes of Map Reduce jobs. We further show that delay intolerant users gain even more from our system.

5.Powernap: Eliminating Server Idle Power

Data center power consumption is growing to unprecedented levels: the EPA estimates U.S. data centers will consume 100 billion kilowatt hours annually by 2011. Much of this energy is wasted in idle systems: in typical deployments, server utilization is below 30%, but idle servers still consume 60% of their peak power draw. Typical idle periods though frequent—last seconds or less, confounding simple energy-conservation approaches. In this paper, we proposePowerNap,an energy-conservation approach where the entire system transitions rapidly between a high-performance active state and a near zero power idle state in response to instantaneous load. Rather than requiring one-grained power-performance states and complex load-proportional operation from each system component, Powernap instead calls for minimizing idle power and transition time, which are simpler optimization goals. Based on the Powernap concept, we develop requirements and outline mechanisms to eliminate idle power waste in enterprise blade servers. Because Powernap operates in lowefficiency regions of current blade center power supplies, we introduce the Redundant Array for Inexpensive Load Sharing (RAILS), a power provisioning approach that provides high conversion efficiency across the entire range of PowerNap's power demands. Using utilization traces collected from enterprise-scale commercial deployments, we demonstrate that, together, Powernap and RAILS reduce average server power consumption by 74%.

6.Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage

Reducing the energy consumption of PCs is becoming increasingly important with rising energy costs and environmental concerns. Sleep states such as S3 (suspend to RAM) save energy, but are often not appropriate because ongoing networking tasks, such as accepting remote desktop logins or performing background file transfers, must be supported. In this paper we present Somniloquy, an architecture that augments network interfaces to allow PCs in S3 to be responsive to network traffic. We show that many applications, such as remote desktop and VoIP, can be supported without application-specific code in the augmented network interface by using application-level wakeup triggers. A further class of applications, such as instant messaging and peer-to-peer file sharing, can be supported with modest processing and memory resources in the network interface. Experiments using our prototype omniloquy implementation, a USB-based network interface, demonstrates energy savings of 60% to 80% in most commonly occuring scenarios. This translates to significant cost savings for PC users.

7.Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services

Energy consumption in hosting Internet services is becoming a pressing issue as these services scale up. Dynamic server provisioning techniques are effective in turning off unnecessary servers to save energy. Such techniques, mostly studied for request-response services, face challenges in the context of connection servers that host a large number of long-lived TCP connections. In this paper, we characterize unique properties, performance, and power models of connection servers, based on a real data trace collected from the deployed Windows Live Messenger. Using the models, we design server provisioning and load dispatching algorithms and study subtle interactions between them. We show that our algorithms can save a significant amount of energy without sacrificing user experiences.

MODULE DESCRIPTION

1. User Upload
2. Allocation of Data center
3. Server workload Management
4. Data overload Performance
5. Resource Allocation

Users Upload

A user can select a file from local machine. When the form is submitted (perhaps together with other form data), the file is uploaded to the web server. The class can process multiple files selected with the file or text form. The descriptions are picked from the value of a form text field that is submitted with the file field data.

Allocation of Data Center

A data center (sometimes spelled datacenter) is a centralized repository, either physical or virtual, for the storage, management, and dissemination of data and information organized around a particular body of knowledge or pertaining to a particular business. The user's uploaded file is stored in the cloud server as compress data form. The cloud server sets the space for users and also it gets unlimited data from the users.

Server workload Management

In server workload management, the workload is the amount of processing that the users have been given to do at a given time. The workload consists of some amount of application programming running in the server and usually some number of users connected to and interacting with the server's applications. The workload management also evaluates the user's data and data's space.

Data overload Performance

The cloud server monitoring the server's space for storing the data from the user. The capacity of a PM (Server) should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its VMs. We should keep the utilization of PMs low to reduce the possibility of overload in case the resource needs of VMs increase later. This indicates that the server is overloaded and hence some VMs running on it should be migrated away.

Resource Allocation

An allocation is a group of allocated resources with a certain percentage of resources guaranteed. When an organization virtual datacenter is created using the Allocation, a dynamic resource is instantiated. This resource group automatically adjusts available resources as new workloads are powered on based on the values specified within the virtual datacenter. Each value has a direct impact on how the related resource pool dynamically changes as new virtual machines are deployed. We develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used.

REFERENCES

[1] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling," Proc. European Conf. Computer Systems (EuroSys '10), 2010.

[2] Y. Agarwal, S. Savage, and R. Gupta, "Sleepserver: A Software-Only Approach for Reducing the Energy Consumption of PCS within Enterprise Environments," Proc. USENIX Ann. Technical Conf., 2010.

[3] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: Fair Scheduling for Distributed Computing Clusters," Proc. ACM Symp. Operating System Principles (SOSP '09), Oct. 2009.

[4] T. Sandworm and K. Lai" Map reduce Optimization Using Regulated Dynamic Prioritization " trans.2011.

[5] D. Meisner, B.T. Gold, and T.F. Wenisch, "Powernap: Eliminating Server Idle Power," Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS '09), 2009.

[6] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, "Somniloquy: Augmenting Network Interfaces to Reduce Pc Energy Usage," Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI '09), 2009.

[7] G. Chen, H. Wenbo, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services," Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI '08), Apr. 2008.

[8] M. Zaharia, A. Konwinski, A.D. Joseph, R.H. Katz, and I. Stoica,"Improving MapReduce Performance in Heterogeneous Environments," Proc. Symp. Operating Systems Design and Implementation (OSDI '08), 2008.

[9] Singh, M. Korupolu, and D. Mohapatra, "Server-Storage Virtualization: Integration and Load Balancing in Data Centers," Proc. ACM/IEEE Conf. Supercomputing, 2008.

[10] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," Proc. IFIP/IEEE Int'l Symp. Integrated Network Management (IM '07), 2007.