Course Project Report of CS 664: *Advanced Operating System*, Spring 2012

Replace the IP routing with Structured Naming and Name-based Routing in Cloud and Data center

Shuai HAO
Department of Computer Science
The College of William & Mary
haos@cs.wm.edu

Qi LUO
Department of Computer Science
The College of William & Mary
qluo@email.wm.edu

Abstract — The emerging model on cloud computing over data center brings new research challenges on the organization and management of resources. Distributed file system doesn't work well in this environment because it fails to provide the cluster computing model. The traditional distributed operating systems also fail since they don't provide the abstraction on multicore and parallel paradigm. To address this problem, one trend is that the cloud and data center should be managed in a single operating system image. In this paper, we follow this argument and explore a new *cross-layer* design on the communication paradigm within the cloud and data center. In our design, all the objects in cloud are organized by the *structured* names, and all messaging and naming are achieved by *name-based intra-domain routing*. With the assumption of a single operating system image managing the cloud and data center, all interactions among the entries within cloud can be implemented by a single system component, which extracts the communication model from IPCs, resource sharing and file system.

1 Introduction

Today's data centers host more and more computing and storage resources to provide public services, such as cloud computing. The emerging data center solutions with clusters of commodity servers accelerate service delivery and deployment, improve quality of service, and reduce risk for new business model. However, the interconnections between a large number of servers and high-volume data transmission require a scalable, efficient and robust underlying network design. Also, the structured storage, virtual machine deployment and distributed computing engine (e.g. *MapReduce*) exacerbate the requirement for networking infrastructure in data centers. Compared with traditional datacenters, the new datacenter uses a homogeneous hardware and system software platform and shares a common systems management layer.

Thus, when we consider operating system-like management layer in data center, it should contain both the software layer for management and abstraction of hardware and a package of tools. In traditional operating system, the single computer needs to play several key roles. It should enable the resource sharing between programs and users and the data sharing between programs. The datacenter also need an OS-like layer for a rising diversity of applications and users [1].

Following this argument that the cloud and data center should have a single operating system image, in this paper we explore a new cross-layer design which combines the naming system and routing in cloud and data center, as a component of emerging data center operating system. In this work, all objects in cloud and data

center, including the processes, resource and files, are assigned structured names, and routed by the name-based intra-domain routing protocol. This method combines all interactions among the objects in cloud and data center into one communication paradigm, and thus all communications like IPCs, resource management, service and file locating can be directly achieve by a single component, the name-based routing.

The rest of this report is organized as follows. In Section 2, we summarize the related work with necessary background. In Section 3, we give an overview the traditional naming scheme. In Section 4, we present the design on structured object-naming and name-based routing in cloud environment and data center. We conclude this paper in Section 5.

2 Related Work and Background

The distributed file system and cluster file system provides the original resource sharing vision for the cloud-like system, like AFS [6] and Pansas [7]. The distributed operating systems like Amoeba [8], Sprite [9], Clouds [10] or also provide the point of view for the cooperation and management of sharing computing and resources in cloud and data center. However, the main design goal of distributed operating systems is to support the time-sharing workloads with the collection of the single-process tasks. Thus, the distributed operating system didn't provide the necessary abstraction on multicore, parallel and cluster computing within modern cloud and data center environment.

The work in [1] specifies high-level views of abstraction and functionality of cluster computing in data center from the operating system perspective. Our work in this report follows this argument that a single operating system image in cloud and data center is necessary to achieve the main functionality the cloud and data center should provide: resource sharing, data sharing, programming abstractions and debugging.

Mesos [11] implements a thin resource sharing layer for the fine-grained sharing between the multiple diverse cluster computing frameworks. It uses the resource offer scheduling mechanism to allow the frameworks to achieve data locality by taking turns reading data, which is stored on each machine. The result of this optimal scheduling shows that the frameworks can meet the goals such as the data locality nearly perfectly. Furthermore, it is easy to be used even if there is only one framework, and it is easy to be developed and experimented with new frameworks.

Another close research to our idea is the factored operating system (fos) [2]. The fos is designed to provide a single OS image instead of fragmented view of cloud resource in multi-core and cloud environment. The fos implements the operating system services by the abstraction of a set of communicating servers, thereby the services in fos are inherently parallel and distributed. We will investigate the messaging and naming scheme of fos in next section. Figure 1 shows an overview of the architecture of fos, a single operating system image in cloud and data center.

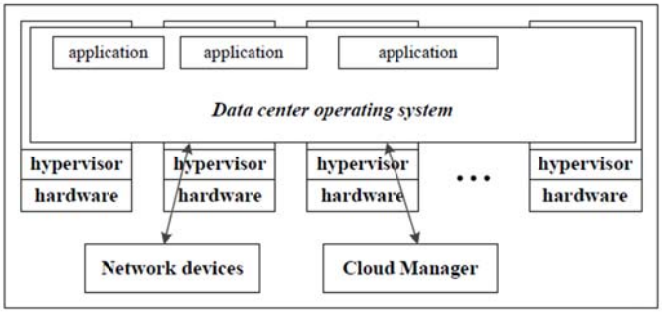

Figure 1. An overview of the single operating system image across the cloud and data center

Our work is also significantly inspired by the *Named Data Networking* (NDN) project [3], a clean-slate design of Internet architecture by naming and routing the named data instead of their locations (addresses). In NDN architecture, the desired data of receiver is identified by a name in an Interest packet, and the data is sent back based on the name of requested data.

There are some interesting design principles which we want to borrow from in the cloud and data center networking:

- In conceptual model, today's applications are typically written in terms of *what* information but Internet only has the knowledge where it is located. Therefore, a middleware layer has to assume the task that maps the entries between applications and networks. The Named Data implements the *what* model and thus removes the middleware and simplifies the network architecture. The cloud and data center can also benefit from *what* model since there are multiple communication paradigms in cloud and the *what* model can intuitively integrate these different communication model by content/name.
- NDN is a replacement of IP layer for today's Internet. Since the cloud and data center is only a local area network, even a flat naming scheme can work in this environment. Thus, it is possible that we explore to replace IP routing into a name-based routing in cloud.
- In NDN, the names are *opaque* to the network. That is, the network devices have no knowledge on the meaning of a name. This may also give the flexibility to the cloud and data center networking: 1) the applications over the cloud can choose their own naming schemes; 2) the naming system can independently evolve from the whole system.

3. Case studies on Naming System

The key component to organize multiple communicating processes in distributed file system and cluster is the naming subsystem. The naming provides the mechanisms of mapping between the logical and physical objects. Also, it provides the methods of locating the data/files in systems.

In this section, we study some typical naming schemes in various classes of distributed computing environment. Andrew File system [6] is a classical distributed file system with scalability and high performance. Also, we investigate the messaging and naming system in fos operating system, which is the closest research to our report.

Naming in Andrew file system

The Andrew File System implements naming and locating by simply adding the symbolic link between local name space and shared name space. The shared name space is identical on all workstations. The cluster servers run the file server process to achieve the file operations like storing and retrieving files in response to the requests from workstations.

The symbolic link apparently cannot work on the cloud and data center since it only provides abstraction on data sharing, but without the model on resource sharing.

fos

In fos, the operating system structure is a form of inter-process communication and synchronization. It provides a simple process-to-process messaging API. In this messaging mechanism, the abstraction will work across several different layers without the concern from the application developer, such that intra-machine communication uses one mechanism while the inter-machine communication uses another.

According to what we have mentioned above, messaging works intra-machine and across the cloud in fos. However, it uses the transport mechanisms that message passes over shared memory. When the messages pass across the cloud, they are sent via shared memory to the local proxy server and then delivered via shared memory on the remote node. Furthermore, the process in fos can register the mailbox by a particular name. The mailboxes are used to receive the message from other processes. The name space is a hierarchical URI just like web address.

In fos, the services are divided into several independent processes, running on different cores within one machine as well as different machines, and these processes are placed into a fleet. All of the servers within the fleet will register under a particular name. When an application messages a given service, the name server will provide a member of the fleet that is best suited for handling the request.

In summary, the fos constructs a single operating system image by dividing the services into the independent processes sequences and providing a process-to-process API to support the messaging and naming. However, the naming scheme is still inefficient: although each process simply needs to obtain the name of the processes it wants to talk, the name server response this lookup by a preliminary flooding in current fos implementation. Although the authors mention that some ideas from P2P networking could be borrowed to improve this mechanism, it introduces additional complexity in naming system.

4. Naming and Routing in Cloud and Data Center

In this section, we present the conceptual model for the architecture design on the structured naming and named-based routing scheme in cloud and data center.

Architecture

The Figure 2 shows a general communication paradigm in traditional data center. The virtual machines with their own operating system run on the hypervisors, and works like a real single machine. The users and applications have to deal with many complex, system-level concerns on the individual virtual machines.

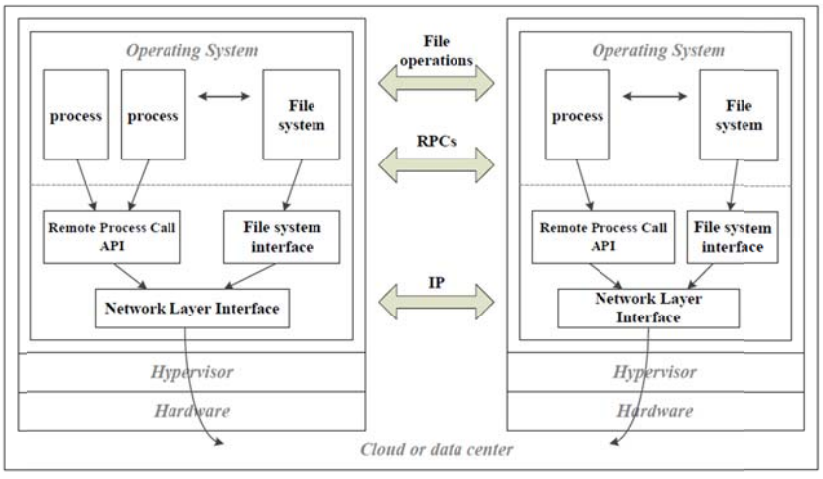

Figure2. A general communication paradigm in data center

Figure 3 illustrate the overview of fos, the single operating system image of cloud. A microkernel runs on each machine (each core if in multicore) and accomplishes a minimum protected messaging layer and a name cache. All other OS functionality and applications execute in the user-level space. As mentioned in the previous section, the Inter-Process Communication in different cores within the same machine is accomplished

via shared memory, and the communications across cloud are achieved by the local proxy network process and the name server.

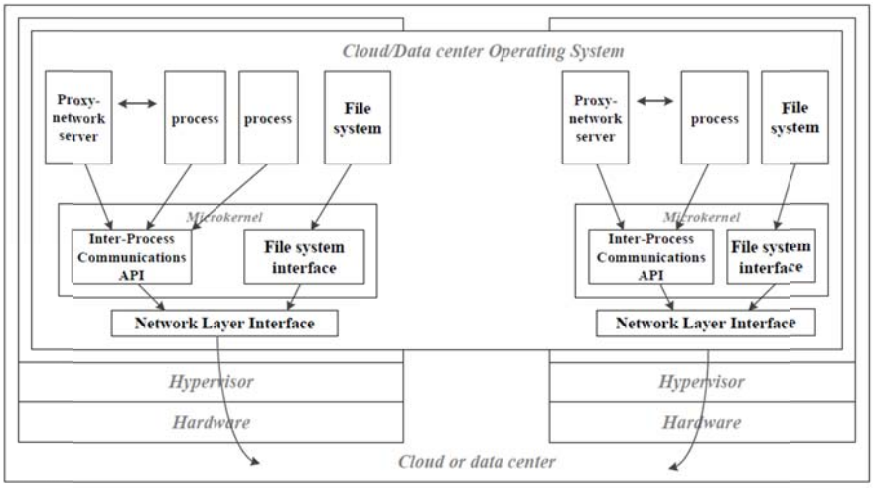

Figure 3. A single OS image communication paradigm in cloud and data center

Like Unix, where all the objects are regarded as the files, in our design, all the objects in cloud and data center are regarded as the named entries, including the processes, services, and files. All objects can be directly routed by the name-based routing protocol. Therefore, a structured URL-like naming is the natural choice for the naming scheme: 1) the structured names should indicate the types of entries; 2) the name can indicate the service of cloud; 3) the name can indicate the entry location in system; 4) the meaning of name remains opaque to the networks.

- Like fos, the processes register their particular names in naming proxy in each machine. The processes providing the same service should register the same name in proxy servers.
- The files stored in data center can be directly assigned the structured name by a hierarchical URL-like name. When one machine is requesting a file in cloud, it does not have to run the name lookup; it only sends the requesting packet with the name of requested file. The naming proxy in the machine containing that files will response the request with the file contents.

The scheme of structured naming and named-data routing can work as an abstraction layer over the tradition IP network layer. In this case, the naming and named-data routing is a pure operating system component over the network stack of data center OS, just like the name-based routing in CDNs. Further, this mechanism also can be used to replace the traditional IP network layer in data center to achieve a real cross-layer with a significant simplicity: the naming system directly serves as the network interface, encapsulates and sends the Name-based packets to the network device, the named-based routers. Figure 4 demonstrates the architecture of this scheme.

An Overview of Name-based Routing

So for, we leave a problem on how the name-based routing can work at cloud and data center. Basically, most of name-based routing protocol refers to the inter-domain routing, using the domain names for routing instead of IP prefixes, or the content-based routing, using in CDNs.

TRIAD [12] is a typical Internet architecture design using the name-based routing. TRIAD architecture defines an explicit *content layer* that provides scalable content routing, caching, naming and routing. Two new

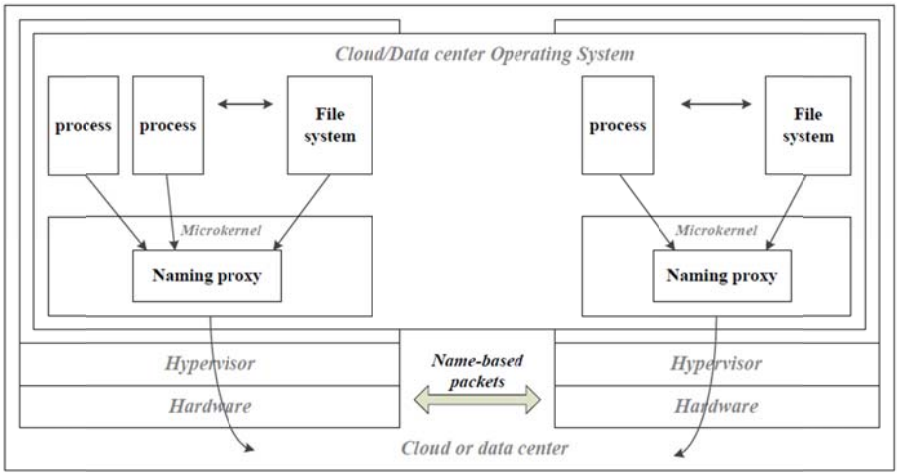

Figure 4. Everything is routable, named entry in cloud and data center

protocols are integrated into the routing: *Internet Name Resolution Protocol* (INRP) and *Name-Based Routing Protocol* (NBRP).

- The Internet Name Resolution Protocol performs name-to-address conversion by routing information of relay nodes. The Name lookups are forwarded towards the authoritative server, and the resulting address is built up by each of the relay nodes involved.
- The Name-Based Routing Protocol (NBRP) is a dynamic mechanism for updating routing information in relay nodes. NBRP distributes name suffix reachability among (and within) address realms.

Obviously, the name-based routing like NBRP cannot be used in the cloud and data center because it is designed for supporting the *content-based* addressing. Also, we want to achieve a direct name routing without the name-address translation. As the introduction in section 2, the NDN architecture gives a feasible solution that can be used in the cloud and data center.

Routing in data center

We use the similar method in NDN to address the name-based routing in data center. The communication is driven by the receiver. To receive data, a consumer sends out a *Name-based packet*, which carries a structured name that identifies the desired objects.

The router remembers the interface from which the request comes in, and then forwards the packet by looking up the name in its Forwarding Information Base (FIB), which can be constructed by the name proxy in each machine exporting its structured name table. This process is similar to a domain announcing its prefix.

Once the requesting packet reaches the machine that has the requested data, a response packet can be sent back which carries the requested objects. This packet traces in reverse the path created by the request packet. Thus, the request packets are routed towards based on the name of objects carried in the packets, and the response packets are returned based on the state information set up by the requesting packet at each router hop.

Because the structured names are opaque to the network, the routers have no knowledge about the meaning of the packet. The Name-based packet can be directly implemented from name proxy process by encapsulating the appropriate name. A public key system can be used to guarantee the data security: the response packets can carry the signature of producer's key.

5. Conclusion

In this report we propose a conceptual model of a new communication paradigm in cloud and data center. In this work, all objects in cloud and data center, including the processes, resource and files, are assigned the structural names, and routed by the name-based intra-domain routing protocol. This method combines all interactions among the objects in cloud and data center into one communication paradigm, and thus all inter-process communications, resource management, service and file locating can be directly achieve by a single communication component in operating system, the naming proxy, with the name-based routing. Specifically, the cloud and data center operating system doesn't have to implement multiple service components for IPCs, naming and distributed file system.

Acknowledgment

This paper is a technical report of student-defined course project for *CS664: Advanced Operating System* in Spring 2012, instructed by Prof. Haining Wang. We want to thank for the instruction and helpful discussion from Prof. Wang.

Reference

[1] M. Zaharia, B. Hindman, A. Konwinski, A. Ghodsi, A. D. Joseph, R. Katz, S. Shenker and I. Stoica, "The Datacenter Needs an Operating System," *HotCloud* 2011

[2] D. Wentzlaff, C. Gruenwald III, N. Beckmann, K. Modzelewski, A. Belay, L. Youseff, J. Miller, and A. Agarwal, "An Operating System for Multicore and Clouds: Mechanisms and Implementation, " *ACM Symposium on Cloud Computing* (*SOCC*), June 2010

[3] Named Data Networking, *http://www.named-data.net/*.

[4] K. Chen, C. Hu, X. Zhang, K. Zheng, Y. Chen, A. V. Vasilakos, "Survey on Routing in Data Centers: Insights and Future Directions, " *IEEE Network*, Vol. 25, No. 4., pp. 6-10. 2011.

[5] J. Rajahalme, M. Särelä, K. Visala, J. Riihijärvi, "On name-based inter-domain routing,"*Journal Computer Networks: The International Journal of Computer and Telecommunications Networking*, Volume 55 Issue 4, March, 2011

[6] J. H. Morris, M. Satyanarayanan, M. H. Conner, J. H. Howard, D. S. Rosenthal, F. Donelson Smith. "Andrew: a distributed personal computing environment", Communication of ACM, 1986

[7] B. Welch, M. Unangst, Z. Abbasi, G. Gibson, B. Mueller, J. Small, J. Zelenka, B. Zhou, "Scalable Performance of the Panasas Parallel File System," *USENIX FAST* '08.

[8] A. S. Tanenbaum, R. van Renesse, H. van Staveren, G. J. Sharp, and S. J. Mullender, "Experiences with the Amoeba distributed operating system," *Commun. ACM*, 33:46–63, December 1990.

[9] J. K. Ousterhout, A. R. Cherenson, F. Douglis, M. N. Nelson, and B. B. Welch, "The Sprite network operating system, " *Computer*, 21:23–36, February 1988.

[10] P. Dasgupta, R. Chen, S. Menon, M. Pearson, R. Ananthanarayanan, U. Ramachandran, M. Ahamad, R. J.LeBlanc, W. Applebe, J. M. Bernabeu-Auban, P. Hutto, M. Khalidi, and C. J. Wileknloh, "The design and implementation of the Clouds distributed operating system, " *USENIX Computing Systems Journal*, 3(1):11–46, 1990.

[11] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. Joseph, R. Katz, S. Shenker, and I. Stoica. "Mesos: A platform for finegrainedresource sharing in the data center, " *NSDI 2011*.

[12] D. R. Cheriton and M. Gritter. "TRIAD: A new next generation Internet architecture," Standford University, March 2000.