HAO Shuai – CSci 664 Advanced Operating System – *Spring 2012*

Paper Review 1

Operating System Features in Data Center and Cloud Computing Environment

Reviewed Papers:

[1] "***The Datacenter Needs an Operating System***",
by Matei Zaharia, Benjamin Hindman, Andy Konwinski, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker and Ion Stoica.
HotCloud 2011

[2] "***An Operating System for Multicore and Clouds: Mechanisms and Implementation***",
by David Wentzlaff, Charles Gruenwald III, Nathan Beckmann, Kevin Modzelewski, Adam Belay, Lamia Youseff, Jason Miller, and Anant Agarwal.
ACM Symposium on Cloud Computing (SOCC), June 2010

I. MAIN IDEA OF PAPERS

The biggest challenge for managing and executing services in Data Center and Cloud Computing environment is that the demands of applications become more dynamic as the number and diversity of applications grow. Therefore, the Data Center and Cloud Computing environment need an operating system-like software layer to implement the functionality of service and management of the applications, programs and data.

The first paper specifies the high-level views of abstraction and functionality of cluster computing in Data Center from the operating system perspective. By abstracting the key roles of traditional operating system, the paper discusses four key functionality of that the Data Center operating system may involve: *resource sharing*, *data sharing*, *programming abstractions* and *debugging*. With the focus on *cross-application concerns*, the authors argue that the new data center OS should provide a common abstraction for different programming paradigm. Therefore, this paper encourages researchers to develop the long-term approach on software infrastructure instead of resolving the problem in an ad-hoc way.

The second paper proposes the design and implementation of a factored operating system, named *fos*, to deal with the dynamic underlying computing infrastructure by a single OS image instead of fragmented view of cloud resource. 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. This paper discusses the challenges of operating system in multicore and Cloud Computing (mainly in the scalability and variability of demand), the components and architecture of *fos* with the case studies of File system and Server fleet, and also presents the preliminary performance measurement for both intra-machine and inter-machine.

II. PROS AND CONS

- *Pros*
  - The first paper provides a thoughtful sight for the new software layer for Data Center from the operating system perspective. It also provides the research directions and focuses for further works.
  - The *fos* specifies an abstraction of single OS image instead of the fragmented view of cloud resource. This abstraction can be built on IaaS or cluster machines. A single OS image is easy to administrate and debug, also can provide the transparent sharing across the cloud and the consistency of process management and recourse allocation.
  - The *fos* is implemented by a clear and elegant hierarchical architecture. A microkernel only provides the messaging protection, name cache, time multiplexing of cores and an API. The execution of all other OS functionalities is in user space.
  - By the design of simple messaging paradigm, the *fos* allows the mechanism of services communicating to be implemented by a common interface in different architecture. Also, the messaging exploits the different transport mechanisms to provide same interface both in intra-machine and inter-machine.

- *Cons*
  - The paper of *fos* didn't describe the mechanism that how to determine which server hosts the process when a massage is sent from a process. It is crucial to understand how the *fos* works.
  - The testbed of only 16 machines didn't produce the convincing results because of the limitations of scale. The scalability mentioned in this paper cannot be evaluated in such small testbed.
  - The experiment of request time of Web Server prototype implemented by *fos* didn't show any improvement from the Linux implementation. The author didn't give any further explanations or discussions for the results.
  - The *fos* complicates the File system implement and introduces the higher overhead. The paper didn't discuss the performance issue on the large-volume data transport.

III. IMPROVEMENT

- The *fos* is implemented as a *Xen* para-virtualization machine OS. As mentioned in the paper, the *fos* don't need a hypervisor, thus it can be implemented by an independent operating system for multicore and Cloud environments. It may bring better performance.
- The key component of the organization of multiple communicating processes is the *Naming* subsystem. Currently the preliminary implementation uses the flooding. The design of *Naming* mechanism can be improved by some technologies in the area of Networking, like the P2P, CDN or NDN.
- The *fos* has implemented a distributed OS services paradigm for cloud. By design new resource abstraction and scheduling mechanism, it may work as an underlying software layer for Data Center OS.