System and Virtualization Management: Standards and the Cloud

Corset: Service-oriented Resource Management System in Linux

22^{nd}-23^{rd} Sept, 2009
Wuhan, China

Dong-Jae, Kang
Contents

1. Goals and Background
2. Architecture and its components
3. Service-oriented Resource Controllers
4. Experimental Evaluation
5. Concluding Remarks
Goal & Problems

- **Goal**
  - Guaranteeing the service QoS or stability in unexpected workload situation

- **Problems**
  - System resources are not enough for many running services and application in a system
  - As the service is more complex, process-unit or system-unit resource management is not enough for guaranteeing the service performance or QoS
  - All services running in a system don’t have same importance
  - Administrator can’t guarantee that the specific service has proper resources in unsettled workload situation
  - System resource management is not easy for administrators
Background
Concept of Corset

- What is ....?

- What can we do with ....?

Automated and intelligent
Service-centric
System Resource Management ....

Mission-critical
Service

Guaranteeing Resources for Service QoS

System Resource Pool
This is whole architecture of CORSET that consists of four subsystems.
Components of Corset (1/2)

- **Resource controller subsystem**
  - May be a kind of resource scheduler or controller
    - Allocate or withdraw each resource to/from the specific service
    - Support predicable or proportional sharing of resources according to service priority
    - CPU, Memory, Disk I/O bandwidth, Network I/O bandwidth and so on

- **Task group management subsystem**
  - Allow administrator to organize a new service with the processes he/she want to include based on PID
    - Create or destroy service
    - Add/delete/move processes to/from specific service
Components of Corset (2/2)

- **Integrated management subsystem**
  - Support dynamic & autonomic reconfiguration according to changeable and unexpected workload
  - Create and provides service-unit resource information to assist service-unit resource management

- **GUI management subsystem**
  - Support convenient and easy environment for system resource management by abstracting the complex interfaces of below blocks
Service-oriented CPU and Memory Controller

- We adapt existing functionalities in linux kernel (about upper kernel-2.6.24)

**CPU**
- CFS scheduler + cgroup framework
- Proportional share

**Memory**
- mem_cgroup + page_cgroup
- Limiting the maximum usage

**Fig.** CPU Control using existing functionality

**Fig.** Memory Control using existing functionality
Service-oriented Resource Controllers (2/3)

- Service-oriented Disk I/O Controller
  - A kind of new device mapper driver
  - Independent of underlying specific I/O schedulers
  - Supported policies
    - Proportional share (weight)
    - Range share (range-bw)

- Fig. Overview of Disk I/O Controller

- http://sourceforge.net/projects/io_band
Service-oriented Resource Controllers (3/3)

- Service-oriented Network Controller
  - TC (traffic control) module already support the limiting or fixed bandwidth
    - Based on IP, port and something like that
  - TC didn’t support the process group based bandwidth control
    - We added assigning, exporting and importing P.G ID in TC

Fig. Overview of Network I/O Controller
Category for Evaluation

- Resource guaranteeing test
  - groupA : groupB : groupC = 1 process : 50 or 100 processes : 50 or 100 processes

- Resource limitation test
  - groupA : groupB : groupC = 1 process : 1 processes : 50 or 100 processes

- Resource isolation test
  - groupA : groupB : groupC = 50 or 100 processes : 50 or 100 processes : 50 or 100 processes

H/W and S/W Specification for test

<table>
<thead>
<tr>
<th>Spec</th>
<th>System</th>
<th>OS</th>
<th>Disk</th>
<th>CPU</th>
<th>Mem</th>
<th>Workloader</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMSUNG</td>
<td>SAMSUNG Smart Server ZSS108</td>
<td>Linux (kernel-2.6.30-rc1)</td>
<td>SAMSUNG SATA-2, 7,200rpm, 80G</td>
<td>2-way, Intel Pentium4 3.4 GHz</td>
<td>2 GB</td>
<td>CPU : ebizzy Mem : ebizzy Disk : fio-1.26 Net : iperf-2.0.4</td>
</tr>
</tbody>
</table>
Experimental Evaluation (Disk I/O BW)

**Fig. I/O bandwidth in resource guaranteeing test (proportional)**

- Group1 : group2 : group3 = 2(1) : 1(100) : 1(100)

**Fig. I/O bandwidth in resource guaranteeing test (range-bw)**

- Group1 : group2 : group3 = 11~12M(1) : 5~6M(100) : 5~6M(100)

**Fig. I/O bandwidth in resource limitation test (proportional)**

- Group1 : group2 : group3 = 1(100) : 2(1) : 2(1)

**Fig. I/O bandwidth in resource isolation test (proportional)**

- Group1 : group2 : group3 = 1(100) : 5(100) : 9(100)
Experimental Evaluation (Network I/O BW)

**Group1 : group2 = 600Mbps(10) : 300Mbps(up to 100)**

![Fig. I/O bandwidth in resource guaranteeing test](image1)

**Group1 : group2 = 700Mbps(10) : 200Mbps(up to 100)**

![Fig. I/O bandwidth in resource limitation test](image2)

**Group1 : group2 : group3 = 10Mbps(50) : 30Mbps(50) : 60Mbps(50)**

![Fig. I/O bandwidth in resource isolation test](image3)
Experimental Evaluation (Memory)

**Group1 : group2 : group3= 10M(1) : 200M(1) : 700M(1)**

**Fig. Memory control in normal test**

**Group1 : group2 : group3= 200M(50) : 400M(1) : 400M(1)**

**Fig. Memory control in resource limitation test**

**Group1 : group2 : group3= 500M(1) : 250M(50) : 250M(50)**

**Fig. Memory control in resource guaranteeing test**

**Group1 : group2 : group3= 500M(50) : 333M(50) : 166M(50)**

**Fig. Memory control in resource isolation test**
Experimental Evaluation (CPU)

Group1 : group2 = 2(1) : 1(200)

Fig. CPU usage in resource guaranteeing test

Group1 : group2 = 1(1) : 2(continuous increase up to 200)

Fig. CPU usage in resource limitation test

Group1 : group2 : group3 : group4 : group5 = 1(100) : 1(100) : 1(100) : 1(100) : 1(100)

Fig. CPU usage in resource isolation test
Features of Corset

- Suggested service-oriented resource management system
  - Allocates the system resources focused on service according to its priority
  - Supports the expectable resources to each service to maintain its QoS
    - Proportional allocation
    - Fixed allocation
    - Range allocation
- Makes the limited system resources to be used efficiently
- Supports the dynamic controls of the system resources

Future works

- Supports the I/O control method for virtual machines


Thank You!

baramsori72@gmail.com
djkang@etri.re.kr