HotChips

High-density Multi-tenant Bare-metal Cloud with Memory Expansion SoC and Power Management

Authors:
- xiantao.zxt@alibaba-inc.com
- zhengxiao.zx@Alibaba-inc.com
- justin.song@alibaba-inc.com
Why Baremetal Cloud and What is X-Dragon?

1. For security and isolation
2. For multi-tenancy and cost efficiency
3. For single-thread performance
4. For interoperability and manageability

X-Dragon: multi-tenant BM-Guests in same Server
Problems

There are VM-based cloud, single-tenant bare-metal cloud and BM-Hive (Multi-tenants bare-metal cloud) in Datacenter

**Problem 1**: VM-based Cloud has non-ignorable virtualization overhead, isolation/security concern and limited single thread performance, but good manageability.

**Problem 2**: Existing bare-metal cloud design for single tenant, lack of manageability and also costly.

---

<table>
<thead>
<tr>
<th>Service</th>
<th>Security</th>
<th>Isolation</th>
<th>Performance</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM-based cloud</td>
<td>Side-channel and Dos attacks because of resource sharing</td>
<td>Weak isolation because of resource sharing</td>
<td>CPU, Memory, and I/O overhead caused by virtualization</td>
<td>Very high density through server over-provisioning</td>
</tr>
<tr>
<td>Single-tenant bare-metal cloud</td>
<td>N/A</td>
<td>Strong isolation due to exclusive access to system</td>
<td>Native performance</td>
<td>Very low density, one user per server, leading to high cost</td>
</tr>
<tr>
<td>X-Dragon</td>
<td>No side-channel or Dos attacks due to hardware-based isolation; Protected hardware resources, particularly the firmware</td>
<td>Strong hardware-based isolation</td>
<td>Native CPU and memory performance; para-virtualized I/O with minor overhead</td>
<td>High, 16 BM-Guests per server at most</td>
</tr>
</tbody>
</table>
X-Dragon High Level View in Cloud

Multiple VM-Guests
- seabios
- vCPU
- Mem
- net
- storage
- vga
- virtio-net
- virtio-blk
- VNC

Qemu/KVM hypervisor

Same cloud Infrastructure: virtual network, cloud storage, etc

Multiple BM-Guests
- bios
- physical CPU
- Mem
- net
- storage
- vga
- virtio-net
- virtio-blk
- VNC

BM-hypervisor

KVM vs X-Dragon

- Same cloud infrastructure
- Same tools to manage
- Both Multi-tenants
- More secure and selectable bare-metal performance
X-Dragon System Architecture

1. Compute Boards + Base Server
2. Hardware implementation of virtio devices
3. Custom backend: BM-Hypervisor
X-Dragon: IO Bond and Backend

**Shadow Ring buffer design**
Transfer data between computing board and backend base server

**BM-Hypervisor design**
Emulate virtio-devices, and connect into existing cloud infrastructure
Evaluation: CPU/Mem/IO performance

- X-Dragon BM-Guest vs Native vs VM: BM-Guests are slightly better performance than VM
- Memory bandwidth: BM-Guests are same as Native. VM 98% of BM-Guests under load
- Network PPS: Same PPS rate, however more implied volatility.
- Latency: Same in application level, longer path then DPDK bypass-kernel testing
- Storage: substantially better than VM from latency and long tail.

Figure 7. CPU performance by SPEC CPU2006
Figure 8. Memory bandwidth by STREAM multi-thread
Figure 9. UDP packet receive rate
Figure 10. UDP and ping latency
Figure 11. Storage I/O latency
Evaluation: Real business

- Nginx
- MariaDB
- Redis

X-Dragon BM guest performs substantially better than the virtualization-based cloud service for the popular applications used in the cloud.
X-Dragon based Infrastructure Enhancement

1. Memory Pool
2. Cloud App Aware Power Management

IaaS TCO optimization and new usage models enabling
Memory Pool

Rack/LocalPool

Switch Fabric

Remote Pool
ROI Analysis

ROI

Lower memory related costs
- Fragments elimination
  - Scheduler – fixed ratio of vCPUs:memory
  - Scheduler – higher density, easy over-provisioning
  - Scheduler – customer constraints
- Tiered media
  - Hot: DDR or HBM
  - Warm: DDR or PMEM
  - Cold: SCM or other
- Operation
  - Easy migration
  - Faster dynamic scaling by pre-warm-up
- IDC
  - Shrunk SKUs
  - Denser deployment
- Opex
  - Lower power & cooling
- Capex

Enable new or easy usage models
- Micro-services
  - Super large memory footprint
  - Rack based deployment
- Support SmartNIC
  - Less local attached memory, less constraint
  - Memory semantic control and optimization
On Compute & Rack

- **Instance**: Instance
- **Hypervisor Memory Mgmt**: Hypervisor Memory Mgmt
- **OS Kernel Mem Mgmt**: OS Kernel Mem Mgmt

**Host**
- Memory allocation
- Page fault handling
- Performance optimization

**CC Controller**
- **MEM Controller**: MEM Controller
- **Cache Line converter**: Cache Line converter
- **NVMe**: NVMe
- **CC bridge**: CC bridge
- **xNIC**: xNIC

**Page Manager**
- **Lookup & order mgmt**: Lookup & order mgmt
- **Warm-up**: Warm-up
- **Prediction & Prefetch**: Prediction & Prefetch
- **Local processing**: Local processing
- **ARM**: ARM
- **ACCL**: ACCL

**Reliable Xfer**
- **NVMe**: NVMe
- **IPS**: IPS

**Buffer / Queue**

**Ethernet / PCIe**

**DDR**

**SSD**

**NVMe**

**PMEM**

**Vendor**

**Alibaba**
## Workloads & Potential Benefits

<table>
<thead>
<tr>
<th>Type</th>
<th>Test</th>
<th>Potential Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Compute</td>
<td>Mid to high utilization</td>
<td>Lower performance, higher density</td>
</tr>
<tr>
<td>Middleware</td>
<td>E-Commerce</td>
<td>Lower performance, higher density</td>
</tr>
<tr>
<td>Micro Services</td>
<td>E-Commerce</td>
<td>Lower performance, higher density</td>
</tr>
<tr>
<td>AI</td>
<td>Ali Native training &amp; inference</td>
<td>Unacceptable for training</td>
</tr>
<tr>
<td>Encryption &amp; Compression</td>
<td>Standard payload pre-/post-processing</td>
<td>Easier to scale out</td>
</tr>
<tr>
<td>Placement &amp; Migration</td>
<td>Large instances</td>
<td>Faster; saving network b/w</td>
</tr>
<tr>
<td>Checkpointing &amp; Mirroring</td>
<td>Cloud based HPC</td>
<td>High performance checkpointing enabled</td>
</tr>
<tr>
<td>NFV</td>
<td>Host gateway</td>
<td>Depends; easier to provision</td>
</tr>
<tr>
<td>Database</td>
<td>In-memory DB</td>
<td>Cost down significantly</td>
</tr>
<tr>
<td>Graph</td>
<td>Large social apps</td>
<td>Cost down significantly; minor programming model change</td>
</tr>
<tr>
<td>Upgrade &amp; Deployment</td>
<td>Patching &amp; initialization</td>
<td>Faster upgrade &amp; composing</td>
</tr>
</tbody>
</table>
Power Management Platform
Highly Available Management

Alibaba Power Agent
• In-Band Power Management
• Out-of-Band Power Management

Server Platform
• Fine granularity power and performance telemetry & control knobs
• In-Band and Out-of-Band Control Channels
Capping & Budgeting

Rack/Node Power Capping

App Driven Power Budgeting

Criteria | Operation |
--- | --- |
A | Rebalance 85%, 3 min |
B | Stop adding instances 100%, 3 min |
C | Super apps flow control 110%, 3 min |
D | Power capping with performance SLA 120%, 5 sec |

High & Low Priority Instances Co-exist on Same CPU

CPU power cap (W)

Frequency (GHz)
## Performance Awareness

<table>
<thead>
<tr>
<th>Type</th>
<th>Target</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>Align w/ apps</td>
<td></td>
</tr>
<tr>
<td>Service delay</td>
<td>1s</td>
<td>Local</td>
</tr>
<tr>
<td></td>
<td>30s</td>
<td>Global</td>
</tr>
<tr>
<td>Models coverage</td>
<td>Based on spec &amp; test results</td>
<td></td>
</tr>
<tr>
<td>Racks coverage</td>
<td>Based on spec &amp; test results</td>
<td></td>
</tr>
<tr>
<td>Power watermarks</td>
<td>Defined by apps &amp; platform</td>
<td></td>
</tr>
<tr>
<td>Capping accuracy</td>
<td>5%</td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td>Defined by apps</td>
<td>Low priority nodes first capped</td>
</tr>
<tr>
<td>Fmin</td>
<td>Defined by apps</td>
<td>Anytime higher than Fmin</td>
</tr>
<tr>
<td>Granularity</td>
<td>By core (CPU), rank (mem), link (IO) and device (storage)</td>
<td></td>
</tr>
<tr>
<td>Capping - DVFS</td>
<td>Minimal performance impact</td>
<td>Defined by apps</td>
</tr>
<tr>
<td>Capping - CCx</td>
<td>Minimal latency impact</td>
<td>Defined by apps</td>
</tr>
<tr>
<td>In-Band</td>
<td>supported</td>
<td></td>
</tr>
<tr>
<td>Out-of-Band</td>
<td>Partially supported</td>
<td></td>
</tr>
<tr>
<td>Thermal watermarks</td>
<td>Defined by apps and platform</td>
<td></td>
</tr>
<tr>
<td>Failover</td>
<td>Unconditional capping, autonomous capping, or SS</td>
<td></td>
</tr>
</tbody>
</table>

Identify IDC, server and app control knobs with least performance impact