Adaptive Memory System over Ethernet

Jun Suzuki, Teruyuki Baba, Yoichi Hidaka†, Junichi Higuchi, Nobuharu Kami, Satoshi Uchida††, Masahiko Takahashi, Tomoyoshi Sugawara, and Takashi Yoshikawa

System Platforms Research Laboratories, NEC Corporation
†IP Network Division, NEC Corporation
††2nd Computer Software Division, NEC Corporation
Memory Scalability in Cloud Computing

Computer memory is limited by individually loaded resources

- Cannot scale depending on service requirements
- Service performance limited by memory
- Slow block I/O devices

Needs for scaling memory beyond individually loaded amount
Resource Should Simultaneously be High-Performance and Networked

**High-Performance**

- Large throughput, low latency
- Avoid firmware process and memory copy to transfer data

**Networked**

- Resource share among multiple computers
- Ease of management
Related Works

(A) Intel Turbo Memory

- PCIe flash device for disk cache
- Device driver between OS and disk driver


(B) Remote Page Swap

- Using memory of next machine with swapping
- Standard interconnection, e.g., Ethernet

Our Method: Ethernet-Attached SSD as High-Speed Swap Device

High-Performance **AND** Resource Share

- Standard Ethernet, PCIe SSD

PCle DMA over Ethernet

- PCIe DMA over Ethernet
  - No Firmware Process
  - No Memory Copy

- Extending PCIe Tree over Ethernet
  - PCIe packet encapsulation into Ethernet frames
  - Ethernet region is PCIe switch

- High-Speed Ethernet Transport [1]
  - Delay-based congestion control
  - < 8.5% of TCP-based delay

Hot-Plug and Remove

SSDs Assigned to Computer with VLAN Grouping

- Adaptive assignment using system manager
- PCIe-standard hot-plug and remove
Evaluations

- Block I/O Performance of Ethernet-Attached SSD
- System Evaluation: In-Memory DB
Evaluation Setups

**Proposal**

(a) ExpEther

**Conventional**

(b) iSCSI

(c) iSCSI with TOE
Block I/O Performance (IOPS) of Ethernet-Attached SSD

Read Close to Host I/O Slot, Write Twice of TOE iSCSI

<table>
<thead>
<tr>
<th></th>
<th>Host I/O Slot</th>
<th>ExpEther</th>
<th>iSCSI w/ TOE</th>
<th>iSCSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ran. Read</td>
<td>100</td>
<td>92</td>
<td>50</td>
<td>14</td>
</tr>
<tr>
<td>Ran. Write</td>
<td>100</td>
<td>74</td>
<td>42</td>
<td>14</td>
</tr>
<tr>
<td>Ran. Read w/ Switch</td>
<td>100</td>
<td>91</td>
<td>46</td>
<td>14</td>
</tr>
<tr>
<td>Ran. Write w/ Switch</td>
<td>100</td>
<td>68</td>
<td>39</td>
<td>14</td>
</tr>
</tbody>
</table>

(a) Random Read IOPS

(b) Random Write IOPS
Write IOPS Overhead and Its Solution

Number of SSD’s outstanding read request limited by its implementation

Increasing number of requests enhances performance close to host I/O slot
System Evaluation: In-Memory Database

Placing RDB File on Ramdisk

RDB: postgresql 8.1
Bench: pgbench (TPC-B-like)
CPU: Intel Core 2 Quad
OS: CentOS 5.3 (Linux 2.6.18)
Ethernet: 10GbE
SSD: 16-GB Partition of Fusion IO 160 GB (Write Improve Mode)
#Client: 100
Transaction per Client: 1000
Scaling-Up beyond Main Memory

- Maintaining performance when DB files enlarged beyond system memory
- >79% performance of all-in-memory at 4G Mem + ExpEther case
Comparison with Conventional Protocol

Proposal outperforms iSCSI by 139% at best case

[Note] iSCSI with TOE could not be evaluated by software bug. Calculation indicates proposal outperforms it by 21%
Saving CPU Resource for Transaction Processing

High-speed swap saves CPU for user process

![Graph showing CPU utilization and swap consumption.](image)

- Consumed by page swap
- CPU Util. [%]
- Total Size of RDB Files [GB]
- Mem 4G + EE, Mem 2G + EE, Mem 8G, Mem 4G + EE, Mem 2G + EE, Mem 8G

Legend:
- software irq
- hardware irq
- wait
- nice
- system
- user
Conclusion

Adaptive Memory Expansion
with Ethernet-Attached SSD as High-Speed Swap Device

✓ **Standard Components**
  • Standard Ethernet and PCIe SSD
  • No software driver for Ethernet expansion

✓ **High-Performance and Resource Share**
  • PCIe DMA over Ethernet
  • Superior block-io performance than conventional protocol
  • PCIe hot-plug and remove

✓ **Proven System Merits**
  • Maintains database performance beyond system memory
Future Works

Simultaneous Share of SSD among multiple computers
• PCIe I/O virtualization emerges
• Efficient resource utilization
• High-speed data share

Solve Performance Bottleneck of Storage and Database System
• Network storage for system availability
• Performance bottleneck by network storage