Virtual Machine Workloads: The Case for New Benchmarks for NAS

Vasily Tarasov, * Dean Hildebrand, *
Geoff Kuenning, ^ Erez Zadok*

*Stony Brook University, *IBM Research – Almaden, ^Harvey Mudd College
Overview

Clients

Network Attached Storage!

Benchmark!

Storage Companies

FastStorage Inc.
NeverFail Corp.
TroubleFree LLC
Overview

But not the benchmarks!
Overview

Create **benchmarks for NAS** that accurately generate workloads originating in **Virtual Environments**

*But not the benchmarks!*
Outline

1. Background
2. Experimental Setup
3. Workload Characterization
4. Benchmark Creation
5. Benchmark Accuracy
6. Conclusions and Future Work
Data Access Options for VMs

File system interface in VMs

1. Distributed File System client in guest OS

VMWare Workstation with VDI on Ext3

(VDI – Virtual Disk Image)

2. Emulated block device + Guest OS on-disk FS

a. VDI on DAS
b. VDI on SAN
c. VDI on NAS
d. Pass-through to DAS/SAN

VMWare ESXi with VDI on VMFS

VMWare ESXi with VDI on a filer

VMWare ESXi with SAN
Trends

- **Shipped storage capacity**
  - Network-based Storage
  - Growth
    - NAS – 60%
    - SAN – 23%
  - [IDC 2010] (by 2014)

- **x86 server virtualization**
  - Virtualized servers
  - Non-virtualized servers
  - [Gartner 2010] (by 2014)
Virtual Machine Workloads: The Case for New Benchmarks for NAS

- Shipped storage capacity
  - NAS: 60%
  - SAN: 23%

- x86 server virtualization

- Trends
  - [IDC 2010] 90%
  - [Gartner 2010]

VMs on NAS

Network Attached Storage

- Direct-attached storage
- Non-virtualized servers
- 70%
Virtual Machine Workloads: The Case for New Benchmarks for NAS

Reordering, Merging, Splitting, …

Read/Write sizes, Attribute caching, …

Network

Applications
Virtual File System
On-Disk File System
Block Layer
Controller Driver

Controller Emulator
NFS Client

NFS Server
Virtual File System
On-Disk File System
Block Layer
Controller Driver

Guest OS

Mapping, Caching, Coalescing, …

Hypervisor & VM

Various Proposed Optimizations

Host OS
VM-NAS I/O Stack

1. Deep I/O stack
2. Requests change significantly
Outline

1. Background
2. Experimental Setup
3. Workload Characterization
4. Benchmark Creation
5. Benchmark Accuracy
6. Conclusions and Future Work
Virtual Machine Workloads: The Case for New Benchmarks for NAS

Physical vs. Virtual

Physical

- Application
- Physical Machine
  - NFS/SMB
  - W1
  - NAS Appliance
    - GPFS, WAFL, ZFS

Virtual

- Application
- Virtual Machine
  - NFS/SMB
  - W2
  - NAS Appliance
    - GPFS, WAFL, ZFS

Current NAS Benchmarks

W1 ≠ W2

New NAS Benchmarks
Physical vs. Virtual

Physical

- Physical Machine
- Application
  - Physical Machine
  - Physical Machine
  - Physical Machine
- NFS/SMB
- W1
- NAS Appliance
  - GPFS, WAFL, ZFS

Virtual

- Physical Machine
- Application
  - Virtual Machine
  - Application
  - Virtual Machine
  - Application
  - Virtual Machine
- NFS/SMB
- W2
- NAS Appliance
  - GPFS, WAFL, ZFS

Current NAS Benchmarks

<table>
<thead>
<tr>
<th>W1</th>
<th>W2</th>
</tr>
</thead>
<tbody>
<tr>
<td>≠</td>
<td>≠</td>
</tr>
</tbody>
</table>

New NAS Benchmarks

Meta-data ops
SPECsfs2008: 72%
Virtual setup: < 1%
How to Evaluate VM-NAS?

- **Run old benchmarks in VMs**
  - Cumbersome
    - Hypervisors, VMs, OSes, applications
  - Inflexible
    - Many combinations of different workloads
  - Expensive
    - Equipment to run hypervisors and VMs

- **Create new benchmarks**
## Configuration

### Physical Machine
- Hypervisor: VMWare ESXi 5
- NFS client: default settings

### Virtual Machine
- Operating System: Linux/Win
- File system: ext3/NTFS
- I/O scheduler: CFQ
- VM parameters: default

### Application
- Filebench: File, Web, Database server
- Jetstress (Mail server)

### NAS Appliance
- Black box
- Linux
- GPFS tracing facilities

### Back-end File System
Outline

1. Background
2. Experimental Setup
3. Workload Characterization
4. Benchmark Creation
5. Benchmark Accuracy
6. Conclusions and Future Work
## Observations

<table>
<thead>
<tr>
<th>Workload Property</th>
<th>Physical NAS clients</th>
<th>Virtual NAS clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>File and directory count</td>
<td>Many files and directories</td>
<td>Few files per VM</td>
</tr>
<tr>
<td>Directory tree depth</td>
<td>Deep and non-uniform</td>
<td>Shallow and uniform</td>
</tr>
<tr>
<td>File size</td>
<td>Lean towards small files</td>
<td>Multi-gigabyte, but sparse</td>
</tr>
<tr>
<td>Meta-data operations</td>
<td>Many</td>
<td>Almost none</td>
</tr>
<tr>
<td>I/O synchronization</td>
<td>Async and sync</td>
<td>All writes are sync</td>
</tr>
<tr>
<td>In-file randomness</td>
<td>Workload-dependent</td>
<td>Increased randomness</td>
</tr>
<tr>
<td>Cross-file randomness</td>
<td>Workload-dependent</td>
<td>Predictable</td>
</tr>
<tr>
<td>I/O sizes</td>
<td>Workload-dependent</td>
<td>Increased and decreased</td>
</tr>
<tr>
<td>Read-modify-write</td>
<td>Infrequent</td>
<td>More frequent</td>
</tr>
<tr>
<td>Think time</td>
<td>Workload-dependent</td>
<td>Increased</td>
</tr>
</tbody>
</table>
Workload Features

- Read/write ratio
- I/O Size distribution
- Jump distance distribution
  - LBA distance between two consecutive requests
- Offset reuse

![Bar chart showing read and write percentages for different server types](chart_image)
I/O Size, Jump Distance, Offset Popularity

- **Web Server**
- **Database Server**
- **JetStress**

**I/O Size (KB)**

- Reads
- Writes

**Jump Distance (GB)**

- Reads
- Writes

**Number of accesses**

- Reads
- Writes
Outline

1. Background
2. Experimental Setup
3. Workload Characterization
4. Benchmark Creation
5. Benchmark Accuracy
6. Conclusions and Future Work
1. Benchmarks generate stationary workloads – almost **no chunking**
2. Block level traces → NFS traces
   But **block-on-file**.
Multi-VM Declarations

- Define hypervisors and VMs:

  define hypervisor type=esx51 {
    define vm type=fileserver,instances=5 {
      ...
    }
    define vm type=webserver,instances=2 {
      ...
    }
  }
Outline

1. Background
2. Experimental Setup
3. Workload Characterization
4. Benchmark Creation
5. Benchmark Accuracy
6. Conclusions and Future Work
Accuracy Metrics

1. Non-virtualized Benchmark in VM

- Reads/Sec
- Writes/Sec
- Latency
- I/O Utilization
- Queue length
- Request size

2. Virtualized Benchmark

- CPU Utilization
- Memory usage
- Interrupts
- Context switches
- Waiting processes

Monitored NAS parameters:

Compare
Single VM Accuracy (Web-Server)
Multi-VM Accuracy

The graph shows the max error (%) across different metrics such as Reads/Sec, Writes/Sec, Latency, I/O Utilization, Queue Length, Request Size, CPU Utilization, Memory Usage, Interrupts, Context switches, and Waiting processes, as the number of VMs increases from 1 to 8.
Outline

1. Background
2. Experimental Setup
3. Workload Characterization
4. Benchmark Creation
5. Benchmark Accuracy
6. Conclusions and Future Work
Conclusions

- Virtualization causes NAS workloads to change
- Current benchmarks are no longer representative
- Need for VM-oriented benchmarks
  - Trace-based analysis of VM-NAS workloads
  - Trace2Model for NFS traces
  - Created a set of VM-benchmarks
  - Accuracy within 10% on average
Future Work

- Explore other configurations
  - The impact of VM-NAS parameters
  - VM workload classification

- VM-specific workloads
  - snapshotting, boot and update storms

- Emulate I/O request transformations
  - Multi-level trace analysis
  - Higher flexibility