Security and Integrity of a Distributed File Storage in a Virtual Environment

Gaspare Sala[1] Daniele Sgandurra[1] Fabrizio Baiardi[2]

[1]Department of Computer Science, University of Pisa, Italy

[2]Polo G. Marconi - La Spezia, University of Pisa, Italy

SISW Workskop, 2007

Outline

Applications with Distinct Trust Levels

- Secure file sharing among applications with distinct trust levels:
  - Web Services.
  - P2P applications.
- Users share their data only if they receive some assurance about the:
  - Description
  - Enforcement

  of the security policy that controls the sharing.

MAC/MLS Policies

To enable secure file sharing, we need an architecture that:

- Describes and enforces in a centralized way a security policy to handle file requests.
- Forces users to respect their roles when accessing files.
- Supports a large set of MAC or DAC policies.

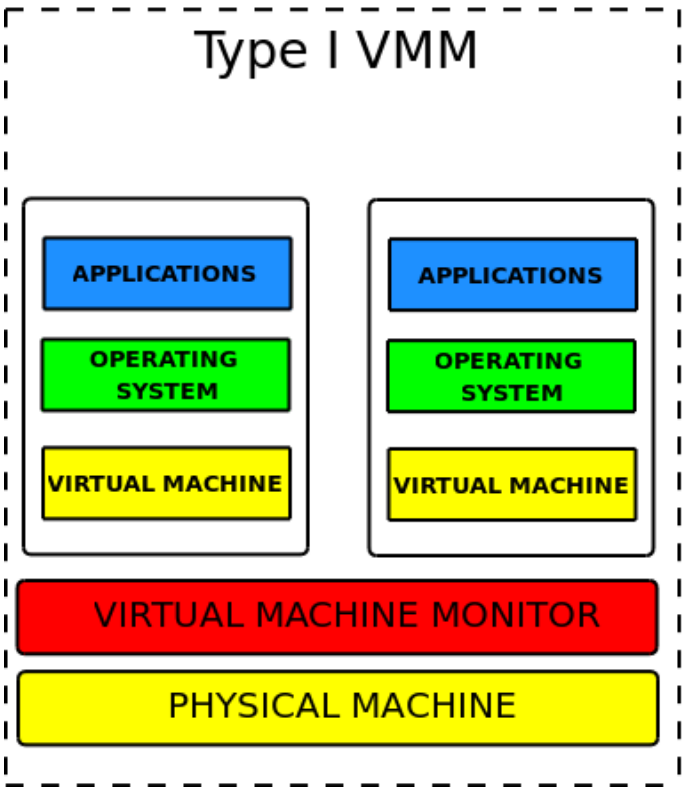
Distributed File System

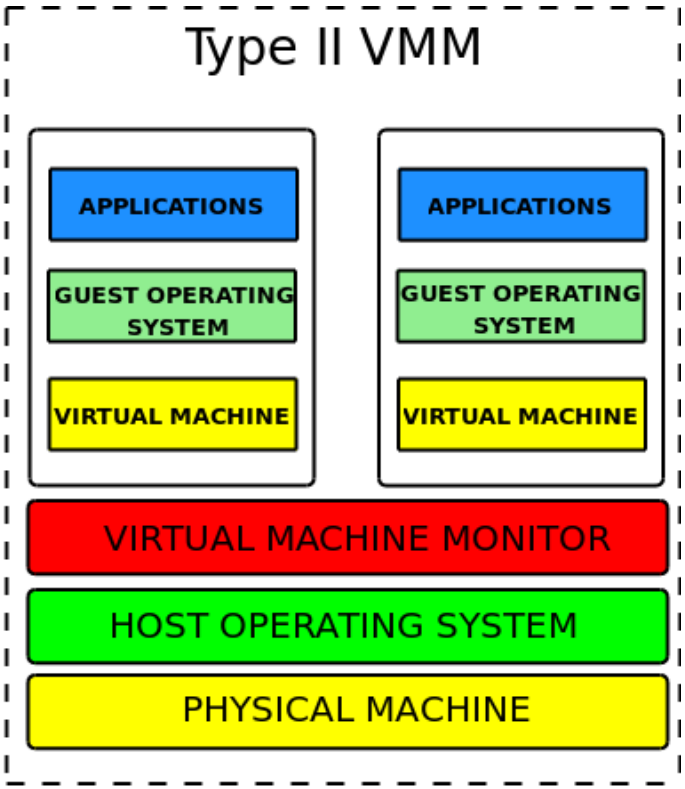
- Client-server architecture to implement a distributed file system.
- Exports to the clients one or more directories of the shared file system.
- Applications access transparently remote shared files.
- Limitations of current solutions: untrusted client user credentials.

Virtualization Technology

- Software emulation of the hardware architecture: Virtual Machines (VMs).
- Benefits:
  1. Confinement among the VMs.
  2. Server consolidation: better resource utilization.
  3. Centralized management: easier administration.
- Widespread usage.

Type I/II VMM


Type I VMM
APPLICATIONS
OPERATING SYSTEM
VIRTUAL MACHINE
APPLICATIONS
OPERATING SYSTEM
VIRTUAL MACHINE
VIRTUAL MACHINE MONITOR
PHYSICAL MACHINE

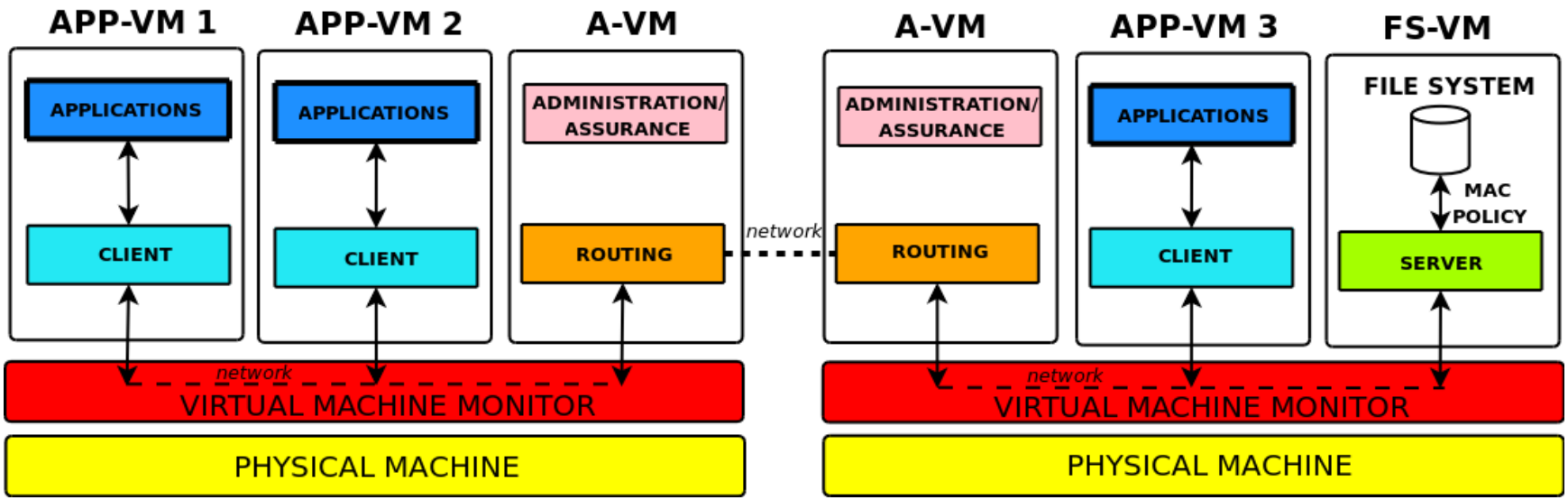

Type II VMM
APPLICATIONS
GUEST OPERATING SYSTEM
VIRTUAL MACHINE
APPLICATIONS
GUEST OPERATING SYSTEM
VIRTUAL MACHINE
VIRTUAL MACHINE MONITOR
HOST OPERATING SYSTEM
PHYSICAL MACHINE

Virtual environment Secure File System

We propose a software architecture for secure file sharing composed of:

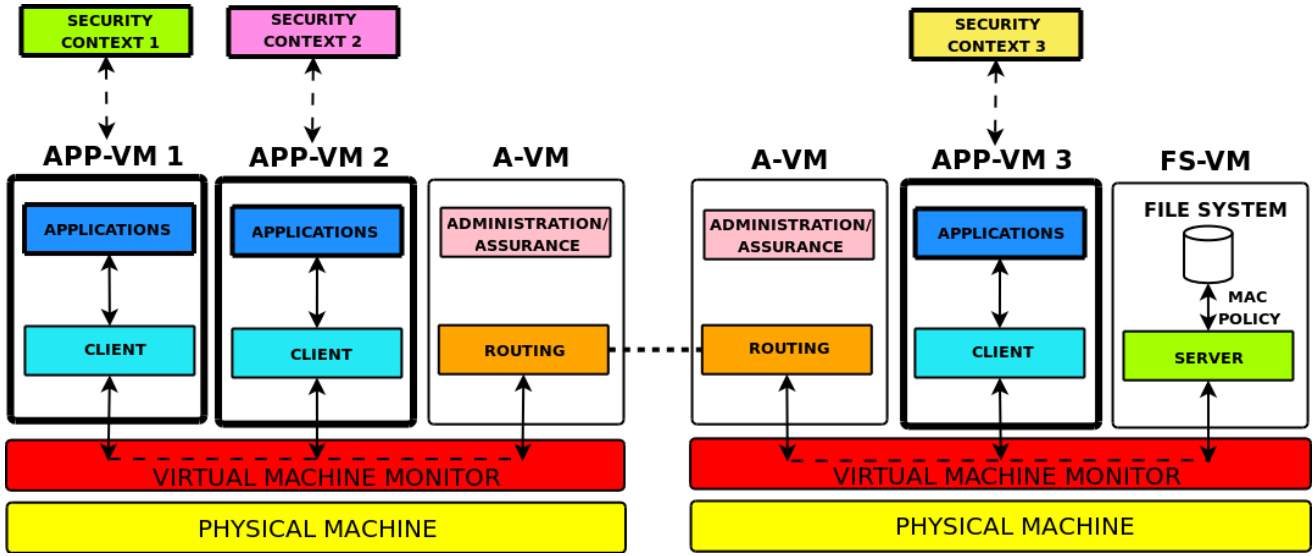
- A network of multiple interconnected virtual machines.
- Three disjoint sets of VMs:
  1. Application-VMs (APP-VMs): each APP-VM runs some application processes.
  2. File System-VMs (FS-VMs): export file systems shared among the application processes.
  3. Administrative-VMs (A-VMs): one for each node, to set up and manage VMs for assurance, routing and administrative tasks.

Architecture



Application VMs (APP-VMs)

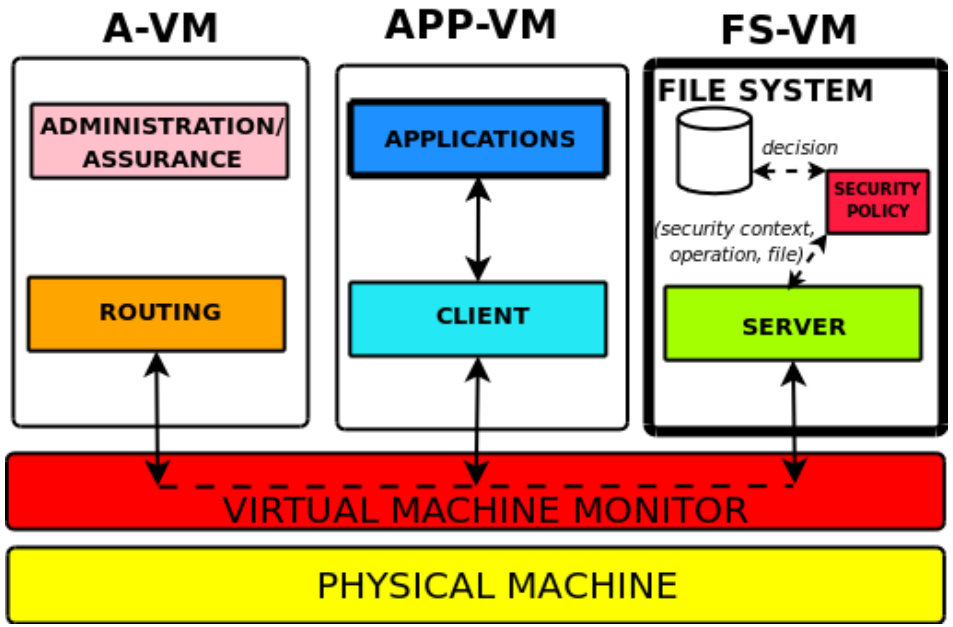
- Run application processes.
- Are labeled with a security context.



UNIVERSITÀ DI PISA

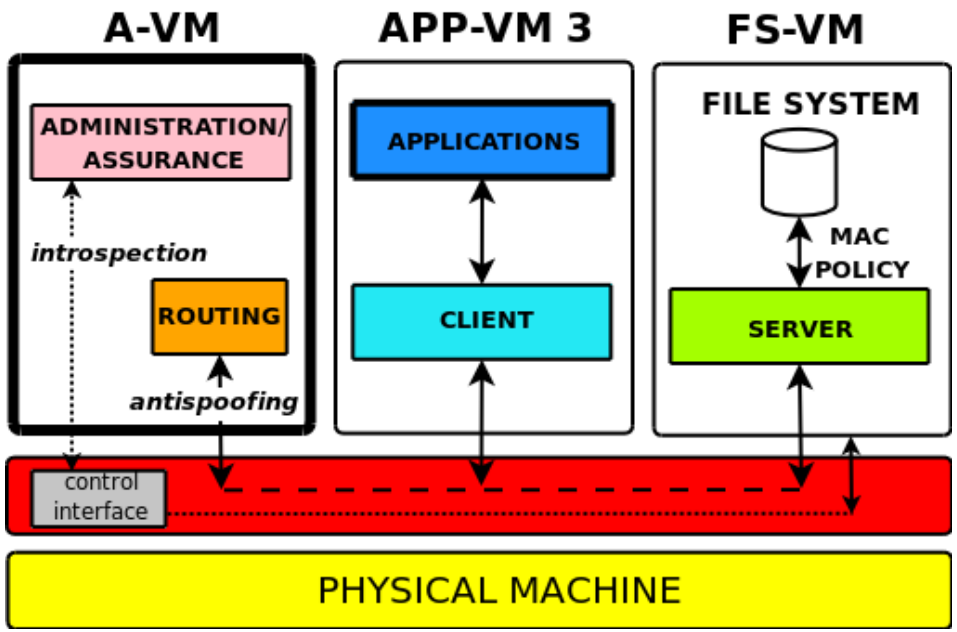
File System VMs (FS-VMs)

- Export file systems.
- Implement MAC policies to control file sharing.



Administrative VMs (A-VMs)

- Protect FS-VM integrity against attacks.
- Implement anti-spoofing techniques to authenticate each file request before routing it.



Threat Model

- VMMs and A-VMs belong to the Trusted Computing Base.
- A malicious application may attacks other ones through shared files.
  - Invalidate data integrity.
  - Contamination through viruses.
- APP-VMs are untrusted: spoofed packets.
- Communications among the physical nodes cannot be forged or spoofed.
- Example: Service Provider using VMs.

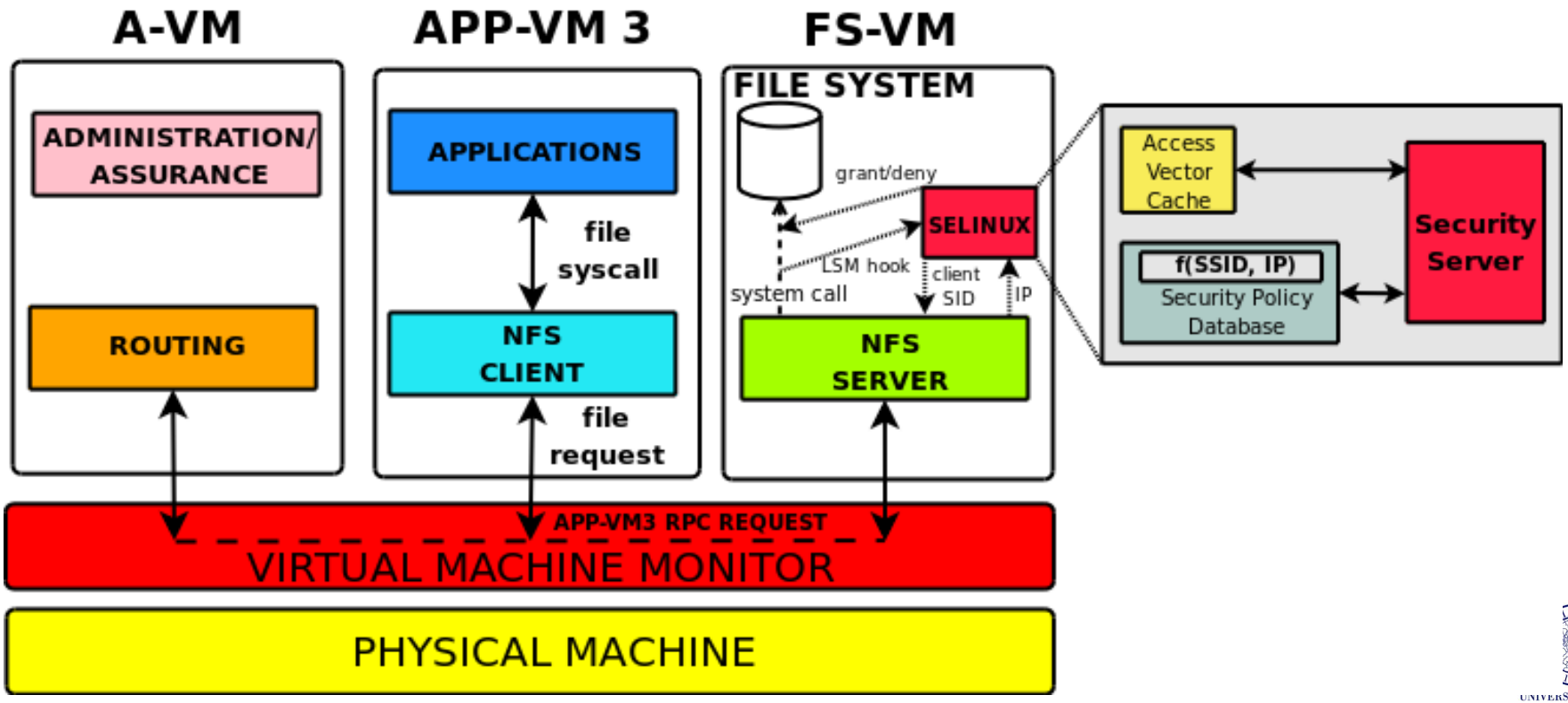
Current Prototype

- Patch to FS-VM Linux Kernel.
- The prototype is based on Xen.
- VSFS exploits NFSv3 service to handle file requests.
- FS-VMs run Security-Enhanced Linux (SELinux):
  1. to support DAC/MAC policies;
  2. to enforce the security policy in a centralized way.

NFS Subject

- Changes to SELinux labeling and access rules:
  - new subject corresponding to the NFS client;
  - definition of all the operations it can invoke.
  - the NFS server acts on behalf of NFS clients.
- VSFS:
  1. Defines a distinct protection domain for each NFS client.
  2. Dynamically pairs the NFS server process with the security context of the NFS client.
- Principle of least privilege.

NFS Request Flow



Assurance

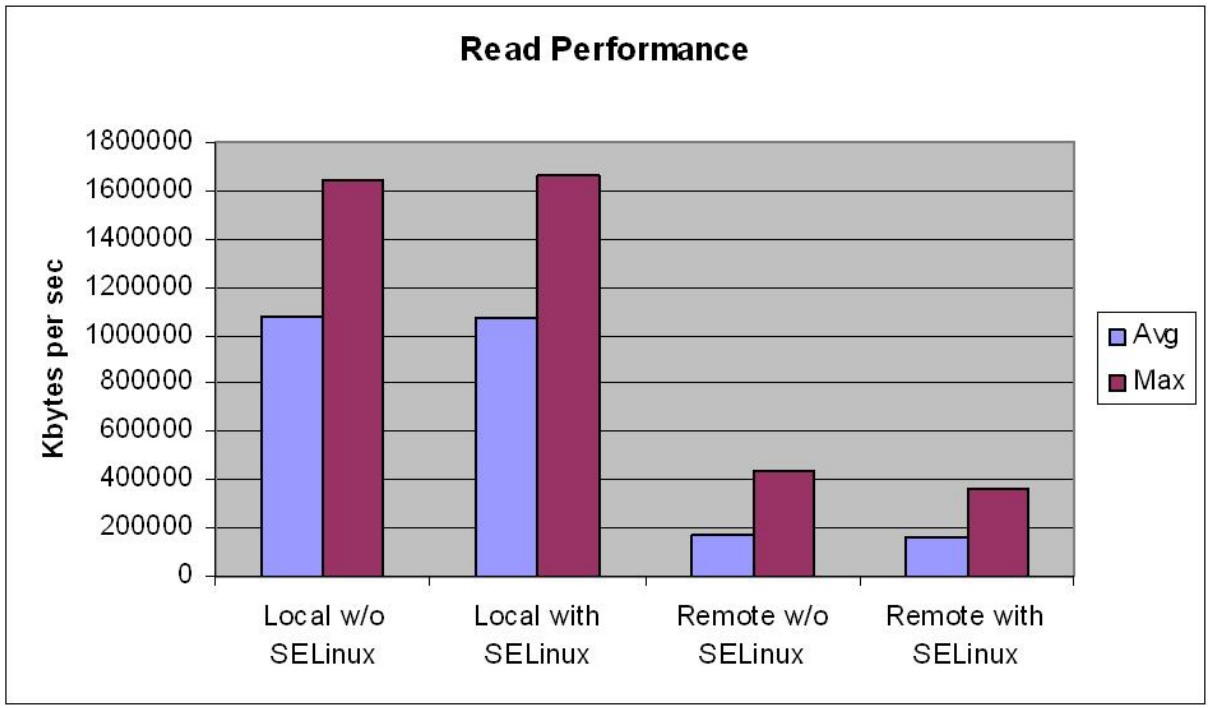
- Virtual Machine Introspection: Standford University.
  - Visibility: access FS-VM's state from a lower level.
  - Robustness: protects FS-VM integrity from an A-VM.
- Anti-spoofing on the Xen virtual bridge:
  - Static IP addresses bound to virtual interfaces.
- The AVM can freeze the execution of a VM.

IOzone

- We used the IOzone Filesystem Benchmark to run NFS performance tests.
  - Read/Write test.
- Four cases depending on whether:
  - APP-VM and FS-VM are on the same or different node.
  - Security policy is enforced or disabled.

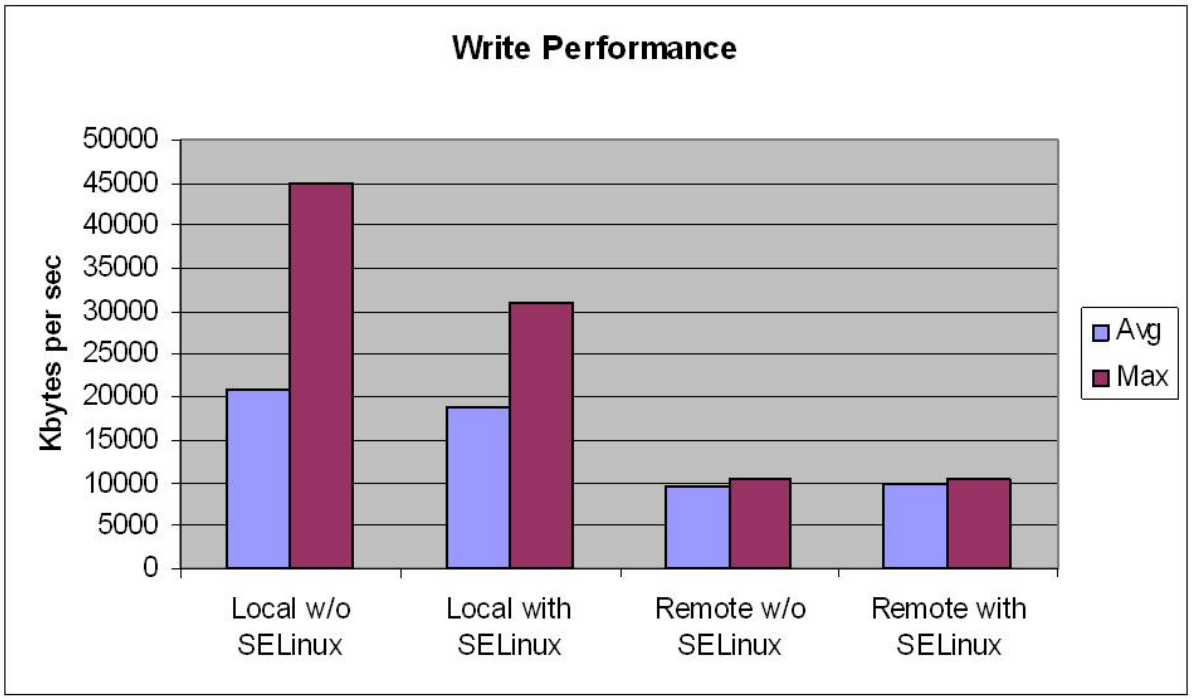
IOzone Read Performance

Overhead is negligible



IOzone Write Performance

Overhead is negligible



Limitations

Current limitations of the prototype:

- No file system encryption.
- Assurance is limited to FS-VMs:
  - attacks to APP-VMs are possible.
- Policy granularity is at the VM level.
- Security policy is static.

Results

- Enforcement of MAC policies on a shared storage:
  - to protect files accessed by applications with distinct trust levels.
- Ability of securely identifying each APP-VM:
  - reliable association of a security context to an APP-VM according to its trust level.
- High assurance of the FS-VM integrity.
- Negligible overhead.

Future Works

- Tainting: track data propagation among users and applications.
- File System encryption.
- Finer-grained security policy: user-ID and NFS client-ID.
  1. Protection domain is a subset of the VM's domain.
  2. Client side authentication.
- Master A-VM: controls and configures the whole network.
  - Ex.: VM migration.
- Support for flexible security policies and MLS.