1 Overview of Course

- Lectures by Prof. Cooperman (at the beginning)
- Readings and Presentations by students (with suggested list of papers to appear)
- Software course project or paper by designing and providing a written report on the software (which can be on a thesis currently being researched). The design will be provided within the first two weeks.

2 Lecture

Def: Virtualization is the interposition on complex systems.

Below is an illustration of how to interpose on with an API:

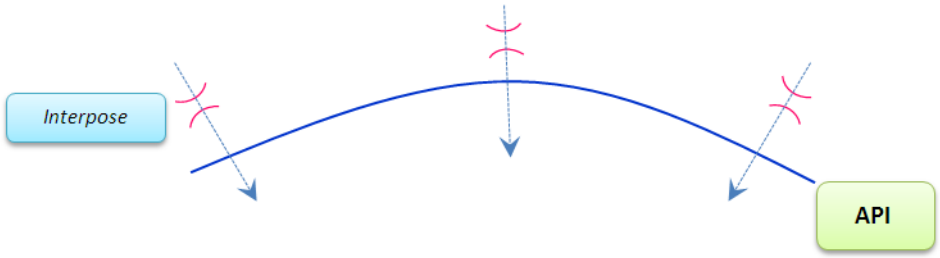

Figure 1: Example of interposing with an API, which can be facilitated via wrapper functions.

Below is an illustration of how to interpose on with the operating system:

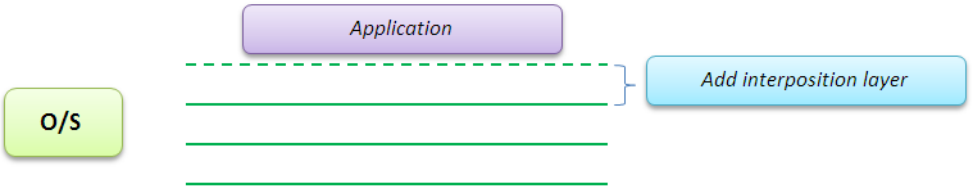

Figure 2: Example of interposing by adding an interposition layer to the operating system.

3 Two Types of Proofs in Computer Science

Mathematical Proof

This is a formal proof with formal verification. Examples of this are the Logic & Computation course for the Semantics of Programming Languages.

Scientific Proof

This is a evidence-based proof, such a programming large software engineering projects.

Science ▶ In science one isolates the mechanism and characterizes each component separately.

Engineering ▶ In engineering one knows something about each component and tries to put them together and construct the program or system.

Evidence-based Proof (Scientific Method)

1. Design simple experiments.
2. Observe the inputs and outputs.
3. Generate hypothesis.
4. Test it.

Example: Testing the Lustre Filesystem

The Lustre filesystem is a complex and scalable filesystem that has been continuously engineered. It can scale to 500 nodes or more but if each node uses more than 1/2 of the RAM, then it can slow down.

Below is a testing criteria:

1. Take one node and try to save 1 GB, 10 GB, 20 GB.
2. Next repeat with more than one node.

This will help develop a Scientific Model (Theory) of what the system is doing.

4 Topics Covered in Course

Microsoft Azure ▶ Microsoft Cloud infrastructure based on Linux over Windows.
dlsym ▶ Dynamic linking, and will compare to static linking.
InfiniBand + RDMA ▶ Will cover general mode.

5 What is the Cloud?

NIST provides five characteristics of the Cloud, but one is most important:

A Cloud is a set of available resources and allows them to grow the number of resources on demand (i.e. elasticity).

Docker is a method by which this can be achieved.

There are two types of Clouds:

Public Cloud ▶ Corporations which have multiple departments sharing resources.
Private Cloud ▶ Google, Amazon, and other providers that people or corporations can purchase resources on, to extend their private Cloud infrastructure.

The Massachusetts Open Cloud (MOC) at Boston University in conjunction with Professor Desnoyers, is an example of a cloud that uses the OpenStack implementation. OpenStack provides a chat with the developers.

Recently Cloud providers (Amazon has 90% of the market) seem to get more expensive with more resources, and more market diversity will help through a bidding process.

5.1 Microsoft Azure

Microsoft Azure is the Linux Operating System running on top of Windows as follows:

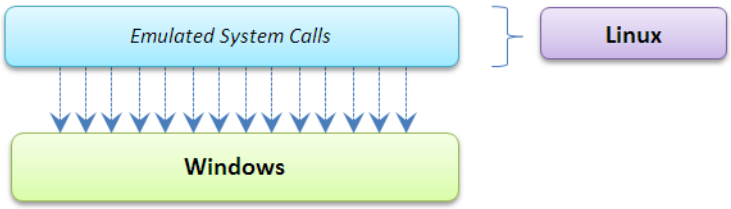

Figure 3: All of the 150 system calls are emulated, and are running on top of the Windows Operating System.

5.2 Dynamic/Static Linking

Static Linking

Containers and Data Centers utilize static linking.

If a file called *a.o* has the function $foo.bar()$ that is being called, the function is being looked in a *Symbol Table* that has the following format:

Symbol Table	
Symbol Name	*Address*

There is also a *Relocation Table* that provides the call site from another function and the symbol, which is dereferenced in the *Symbol Table*:

Relocation Table	
Call Site	*Symbol Name*

5.3 Dynamic/Static Linking (ELF)

The *libc* library is 10s of MB. If one were to statically link for 200 processes with only 412 bytes each, that would lead to a memory allocation of more than 4 GB, which can be optimized if there was sharing of common code.

libc is mostly text which is shared as read-only code.

The old method of storing object data into a library, would be *libc.a* file, which is basically a tar of .o files - most of which are not used by one particular application.

Via virtual memory where each process has its own memory space, then a *libc.so* file can be shared object among multiple .out files (i.e. a.out, b.out, c.out).

Thus a.out file can have multiple shared library objects:

aout: libc.so, libpthread.so, librt.so, ...

The static linking a a.out file would just pull out the .o files that it needed, which would be around the size of a header file.

For interposition the symbol table is needed in order to search in all the files, which can be swapped out as they are hashed.

6 Docker/HPC

Docker containers are built with the Go programming language, which is ***static*** by default.

With HPC systems the same static approach is taken. The key idea of HPC systems is ***throughput*** which would be equivalent to the number of jobs-per-hour (jobs/hour).

Oak Ridge (Tennessee) is the largest supercomputer in the US, and use the amount of electricity of the town of Knoxville. The migration of data takes about a year but would be expensive to run two side-by-side systems for more than that.

With the Cloud the key is elasticity, or on demand resources.

In a Data Center there are no more mainframes but clusters of computers (server farm). In this scenario, reliability is not as important and thus Virtual Machines (VMs) are the latest implementation, which took a long time to develop properly without snapshots. The idea was to have the server die and the recover is made from disk onto another server. Thus no persistence is necessary and thus no snapshots either.

Now we are shifting to having GB containers instead of GB VMs using Linux, where containers are another type of visualization.

7 Containers: Visualization

Containers have: a namespace, control groups (cgroups) and Union File System.

Namespaces

These would be the name of a resources, such as the Process ID or the ethernet port of a network address.

Control Groups (cgroups)

These are a way to share the resources of the computer among different containers.

A container can run as root and set its Process ID to 0 since there is a mapping to the kernel outside to another Process ID.

Kernel	Container
2375	0
4019	1

Containers have the following features:

- Have no snapshots.
- Can be spun up fast.
- Can be provide persistence.

Though using 10,000 containers for one user might be overwhelming to run under one user, this can be achieved on the HPC Cloud.

Union File System

This layer can achieve both RW and RO as the RW layer is above the RO layer and accessed first - which can useful for running a Live OS, or writing a snapshot in the RW layer. It can be viewed as follows with the access starting from the top:

RW
RO

Docker most important feature is that it can modularize the delivery of a process by creating it, placing it into the container and delivering it.

7.1 Orchestration

Orchestration of multiple Docker containers can happen horizontally which has no reliability, or vertically with a virtually shallow layer. A horizontal would need a scheduler which would be more than just cgroups.

The difference between a VM and Docker is that, a VM has visualization performed from the outside-in and thus has access to a kernel.

A Docker on the hand has not real filesystem and thus has no kernel.

Orchestration Systems in a Data Center can orchestrate many VMs or Containers. Examples of these for which we will have invited speakers are: Mesos and Kybernetes.

7.2 dlsym

dlsym makes it easy to interpose. Since the library symbols are read in the order of the library paths from left-to-right (where the left is thought as the highest layer, and right the lowest layer):

libdmtcp.so, libpthread.so, ...

Highest Layer → libdmtcp.so | libpthread.so | ... *Lowest Layer*

If a secondary function definition needs to be accessed down the layers, then the `RTLD_NEXT` parameter can be used to access the second definition of the function `bar`, as follows:

```
dlsym(RTLD_NEXT, "bar")
```

Thus it is possible to interpose on static linking which Professor Cooperman performed by replacing dlsym.

8 Infiniband and RDMA Network

Intel developed OmniPath which would allow it to work on a higher layer. Via Infinityband with RDMA one can write directly to memory between machines on a fast Infiniband network, which providing the parameters for tuning RDMA for specialized implementation.