MyriXen: Message Passing in Xen VMs

Anastassios Nanos, Nectarios Koziris

{ananos,nkoziris}@cslab.ece.ntua.gr

Computing Systems Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens

Motivation

• Cloud Computing is a significant trend, but is mainly used for consolidating service-oriented environments.

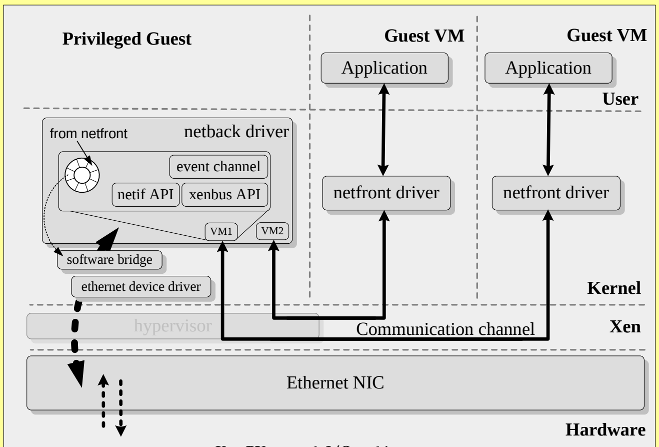
• Bridging the gap between Virtualization techniques and High-Performance Network I/O.

• HPC interconnects provide abstractions that can be exploited in VM execution environments but lack architectural support.

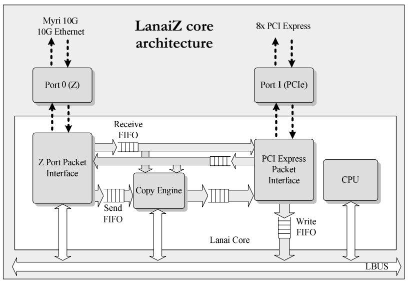
Xen Architecture

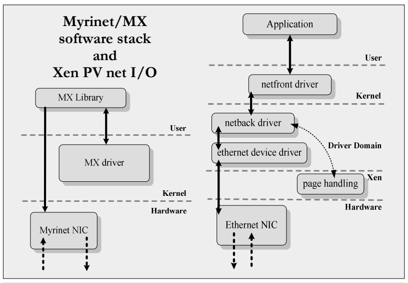
Xen PV network I/O:

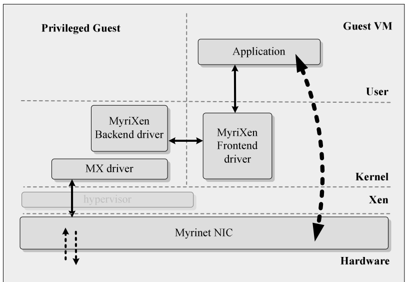
- based on a split driver model (**netfront** / **netback**):
  - exports a generic Ethernet API to kernelspace.
  - driver domain has direct access to hardware, and hosts
    - the hardware-specific driver,
    - the protocol-interface driver and
    - the netback driver
- the **netfront** attaches to the **netback** driver, which in turn attaches to a **dummy network interface** bridged with the physical network interface in software.



Xen PV network I/O architecture







MyriXen Architecture

Myrinet/MX

• An application is granted control of the NI by **mapping** part of the NI memory space into its own virtual memory.

• **User-level communication** is accomplished by mapping parts of the Lanai into the VM address space of an application, bypassing OS abstractions and copies.

• Each of these parts, called **MX endpoints**, acts as an **isolated virtual network interface** at the process level, for the application. It contains an unprotected part, which is mapped to userspace, and a protected, trusted part, which is only accessible by the kernel module and the firmware.

• An endpoint provides an **entry point** to the interconnect's hardware, separated from other processes, with fairness relative to the other endpoints opened on the same NIC.

MyriXen Architecture

• The split driver model used in Xen poses difficulties for user-level direct NIC access in Xen VMs.

• To enable driver domain bypass techniques, we need to let VMs have **direct access to certain NIC resources**.

• The building block of MyriXen is **myriback** which allows **myrifront** to communicate with the **MX core driver**.

• The myrifront driver, communicates with the backend via an **event channel mechanism**.

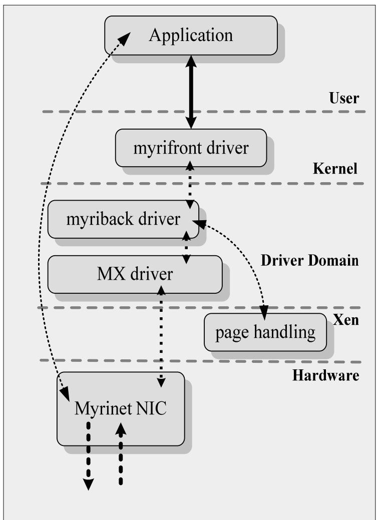
• Contrary to the netfront / netback architecture, **MyriXen utilizes the backend** in conjunction with the core MX driver **to grant pages to the VM user space and install mappings** that can simulate the normal case, while **the netfront driver uses these channels as a data path.**

MyriXen Semantics

To communicate with the network, an application running in a VM needs **privileged access to the NIC**. To provide **isolated, virtualized** access to the Myri-10g NIC we must take into account the following issues:

- **Initialization**
- **Endpoint management**
- **Memory registration**
- **Message Matching**
- **Protection**
- **Address Translation**

Note that initialization, endpoint management and memory registration occur outside the critical path of network-intensive applications.



MyriXen software stack

Challenges

The integration of Virtualization semantics in High Performance Interconnects can be achieved by semantically enhancing both the Myrinet/MX and the Xen Network I/O split driver architecture.

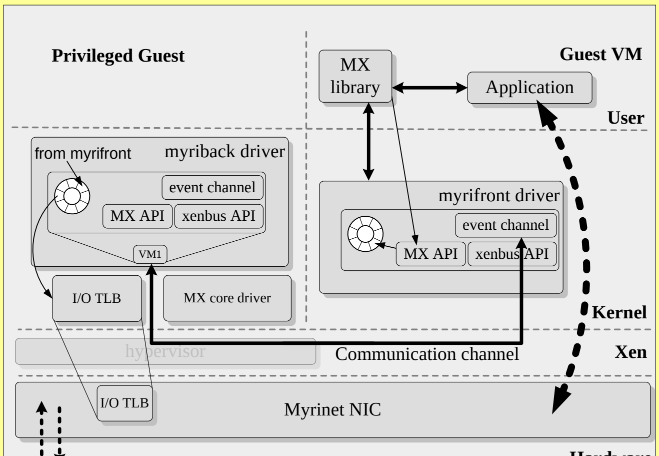
• Integrating **HPC applications** in **Virtualization environments**, such as Cloud Computing infrastructures.

• Deploying **MPI applications over a cluster of VMs** that reside in different VM containers.

• Existing **messaging protocol semantics** can be integrated into **Virtualization platforms**.

Conclusions

MyriXen's architecture makes the best **compromise** between **maintaining Virtualization semantics** and **providing near wire-speed performance for High Performance Myri-10g interconnects**. Our goal is to evaluate our prototype design and present extensive performance measurements of MyriXen in conjunction with latency breakdown analysis on MPI applications running on clusters of VMs. We expect these applications to benefit greatly from **MyriXen's direct data path** and achieve **performance close to native**.



MyriXen network I/O architecture

References

Santos, J.R., Turner, Y., Janakiraman, G., Pratt, I.A.: *Bridging the gap between software and hardware techniques for I/O Virtualization.* In: ATC'08: USENIX 2008 Annual Technical Conference on Annual Technical Conference, Berkeley, CA, USA, USENIX Association (2008) 29–42

Kieran Mansley, Greg Law, D.R.: *Getting 10 Gb/s from Xen: Safe and Fast Device Access from Unprivileged Domains.* In: Euro-Par 2007 Workshops: Parallel Processing. (2007)

Youseff, L., Wolski, R., Gorda, B., Krintz, C.: *Evaluating the Performance Impact of Xen on MPI and Process Execution For HPC Systems.* In: Virtualization Technology in Distributed Computing, 2006. VTDC 2006. First InternationalWorkshop on. (2006)