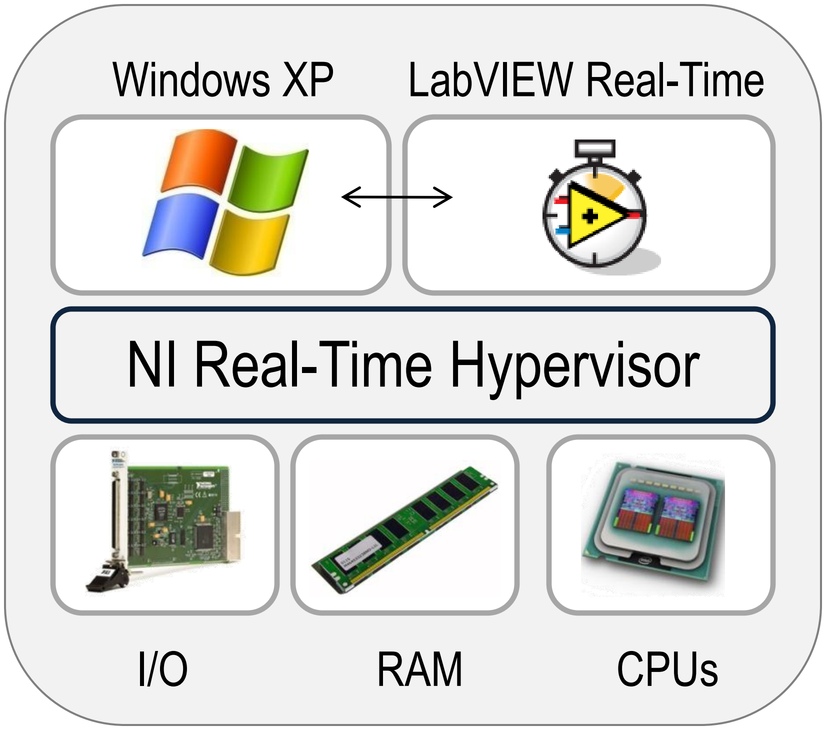
Introduction to the NI Real-Time Hypervisor

Agenda

1) NI Real-Time Hypervisor overview
2) Basics of virtualization technology
3) Configuring and using Real-Time Hypervisor systems
4) Performance and benchmarks
5) Case study: aircraft arrestor system

NI Real-Time Hypervisor Overview

NI Real-Time Hypervisor



- Run NI LabVIEW Real-Time and Windows XP in parallel
- Partition I/O devices, RAM, and CPUs between OSs
- Uses virtualization technology and Intel VT

Benefits of the Real-Time Hypervisor

- Capability: make use of real-time processing and Windows XP services

Applications	Determinism
Graphics	Real-Time I/O
Services	Timing

Benefits of the Real-Time Hypervisor

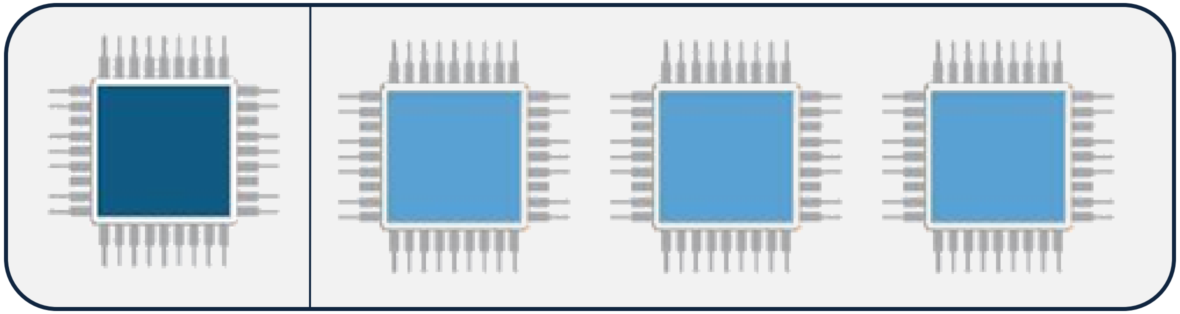
- Consolidation: reduce hardware costs, wiring, and physical footprint

Virtualized System with NI Real-Time Hypervisor

Benefits of the Real-Time Hypervisor

- Efficiency: take advantage of multicore processors effectively

Quad-Core Controller with Virtualization



Windows XP LabVIEW Real-Time

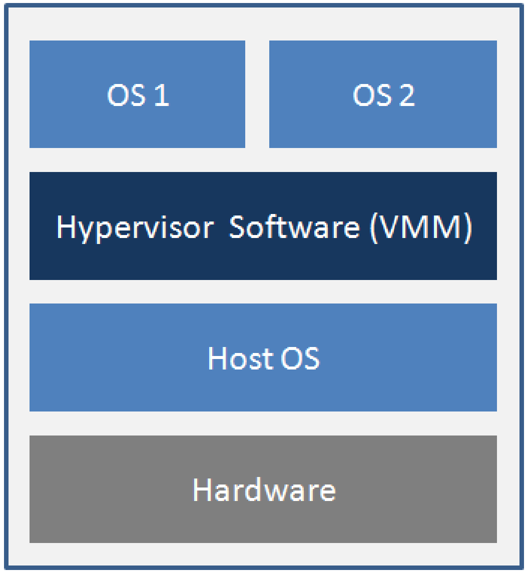
Basics of Virtualization Technology

What Is Virtualization?

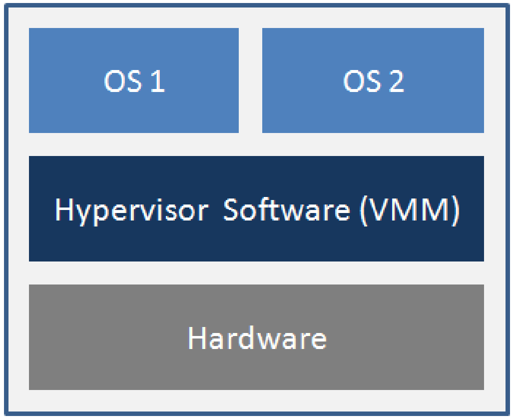
- **The term:** refers to abstraction of OSs from hardware resources
- **In practice:** running multiple OSs simultaneously on a single computer

Virtualization Software Architectures

- Software: virtual machine monitor (VMM) or Hypervisor
- Two main variations: hosted and bare-metal



Hosted (VMWare)

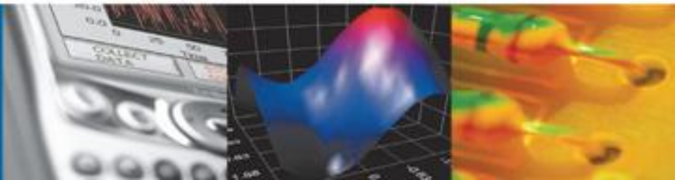


Bare-Metal (NI Real-Time Hypervisor)

How Does Virtualization Software Work?

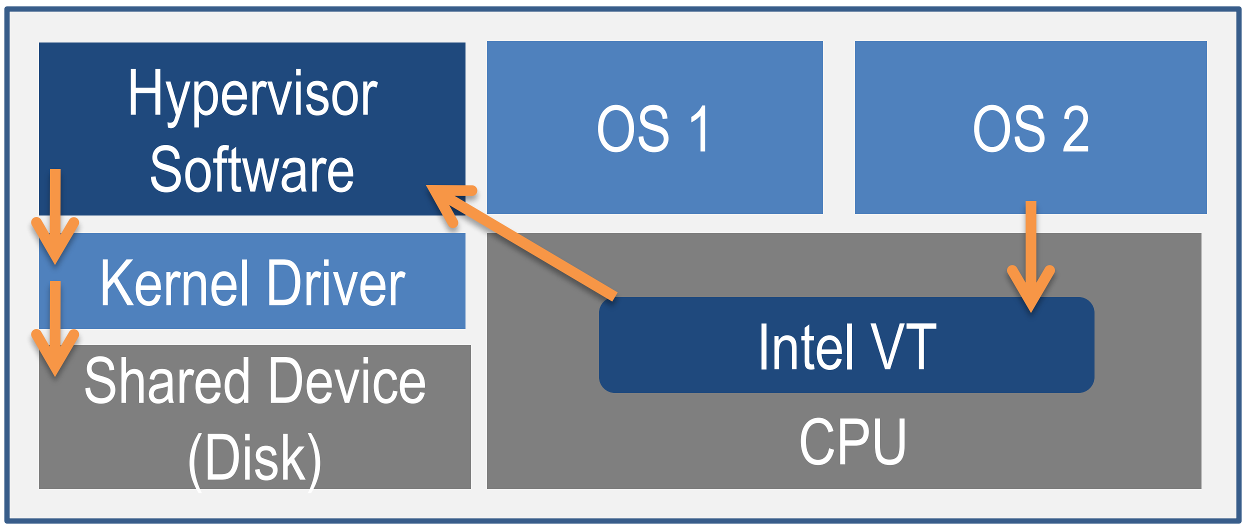
- OSs are “unaware” of being virtualized
- Hypervisor is called only when needed
- Various mechanisms for calling the hypervisor (hardware assist with Intel VT or binary translation)

Hypervisor goal: facilitate simultaneous operation of OSs and protect access to shared system resources



Example: Accessing Shared I/O Devices

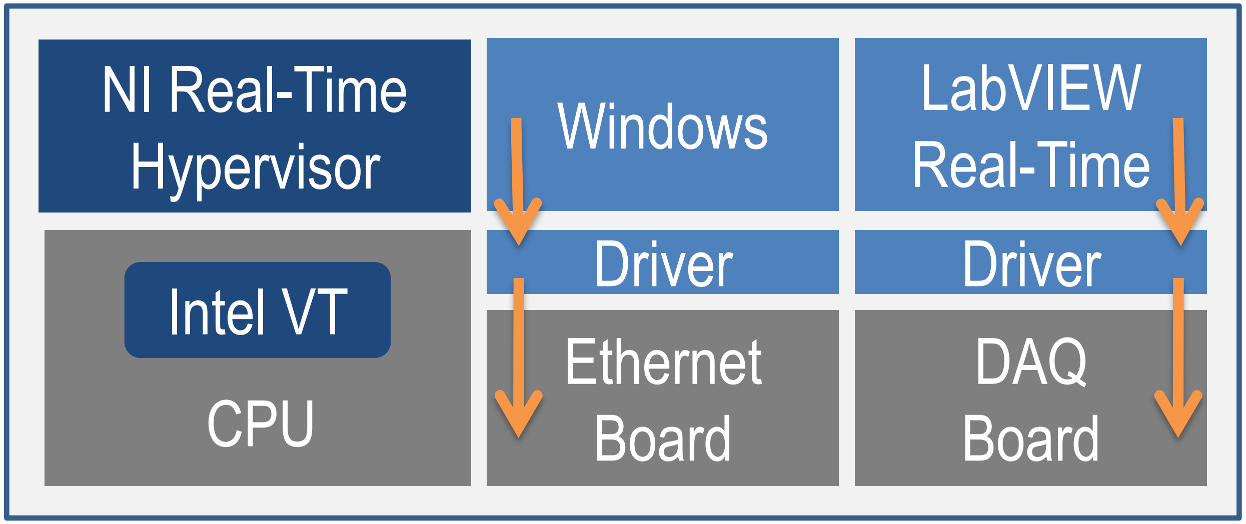
- OS 2 attempts to transfer data to disk
- Processor with Intel VT calls hypervisor
- Hypervisor writes to disk using its own driver



Note: NI Real-Time Hypervisor does <u>not</u> typically do this; devices are partitioned rather than shared

Example: Accessing Partitioned I/O Devices

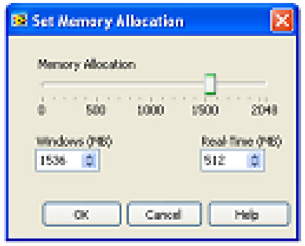
- NI Real-Time Hypervisor allows OSs to communicate directly with partitioned I/O boards



Configuring and Using Real-Time Hypervisor Systems

Using NI Real-Time Hypervisor Systems

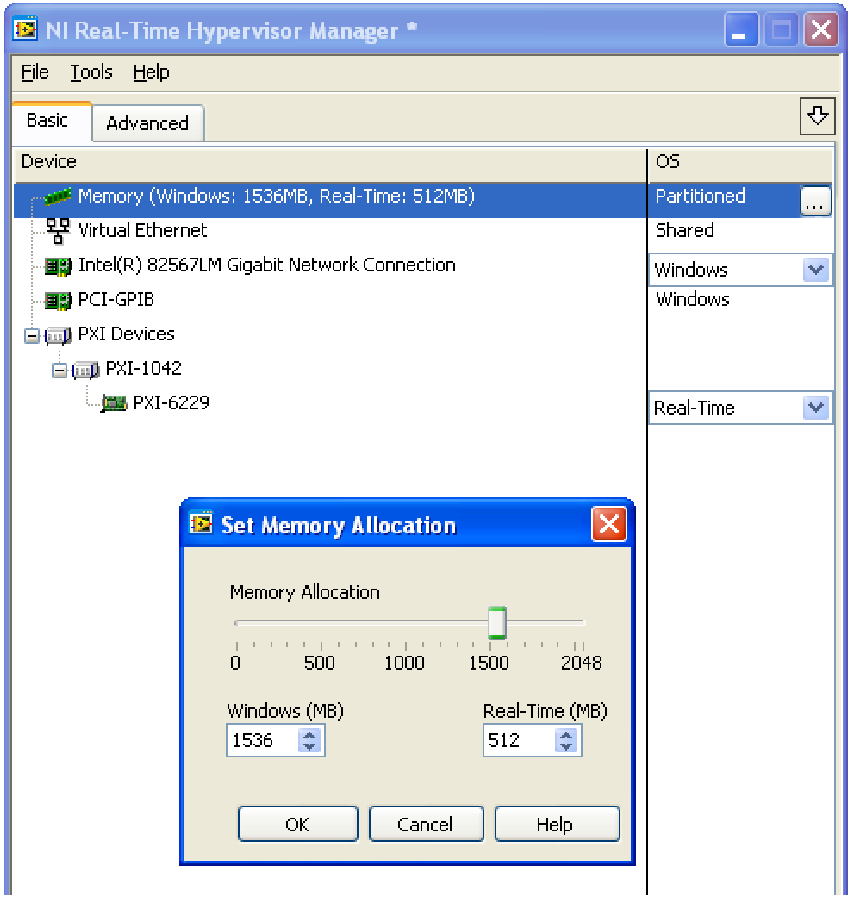
- **Configuration**: NI Real-Time Hypervisor Manager



- **Communication**: virtual Ethernet and virtual console

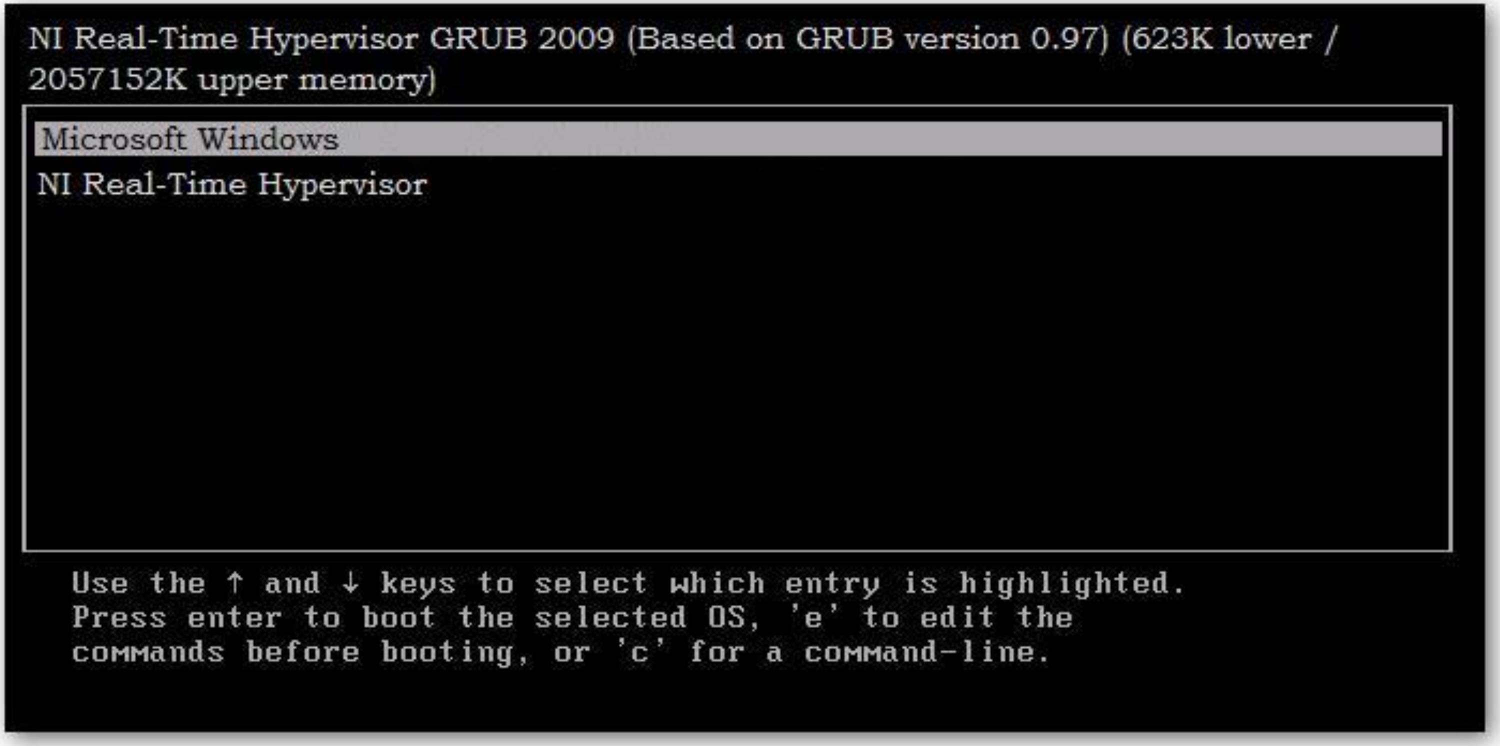
- **Development and Deployment**: similar to traditional real-time systems

Assigning I/O and RAM between OSs

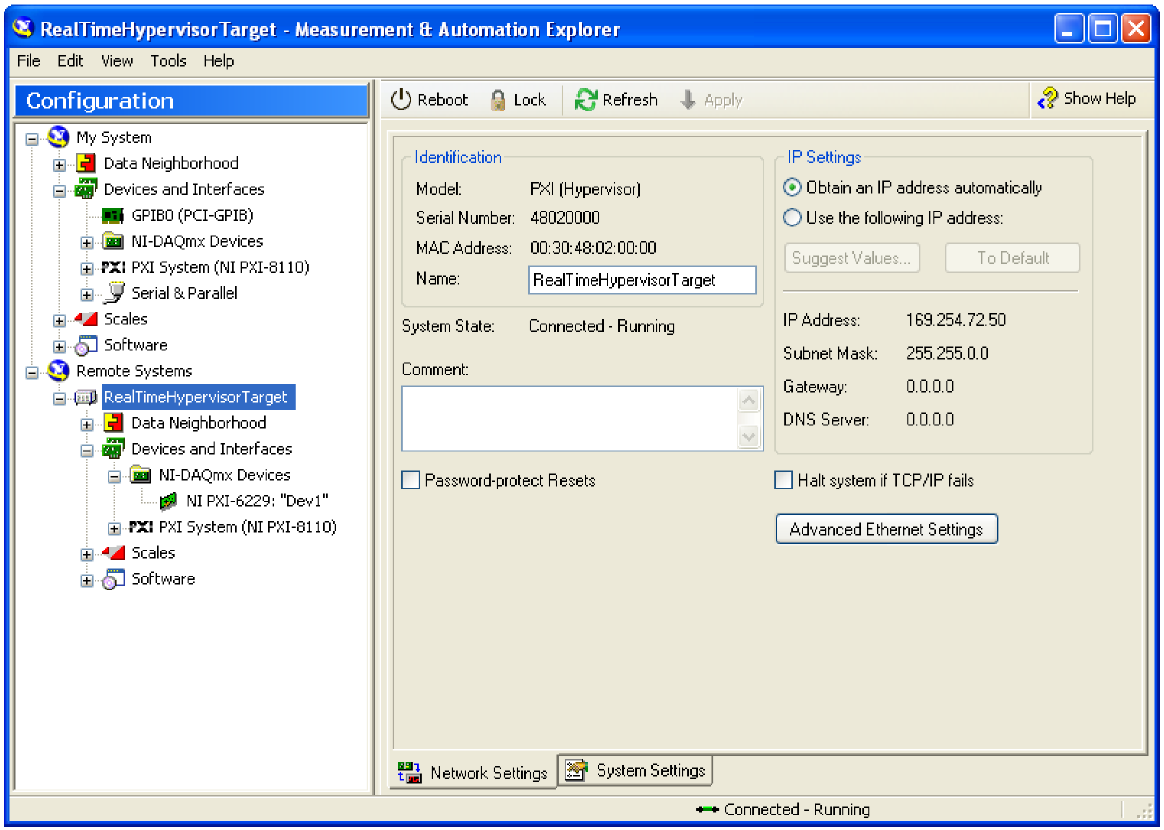


Demo: Configuring a Real-Time Hypervisor System

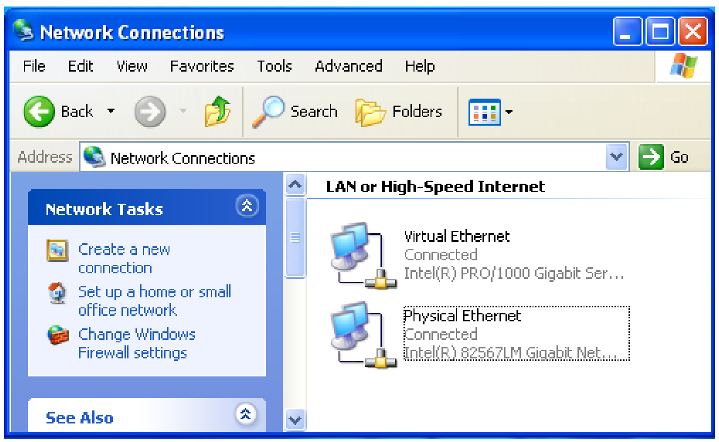
Booting Into the Hypervisor



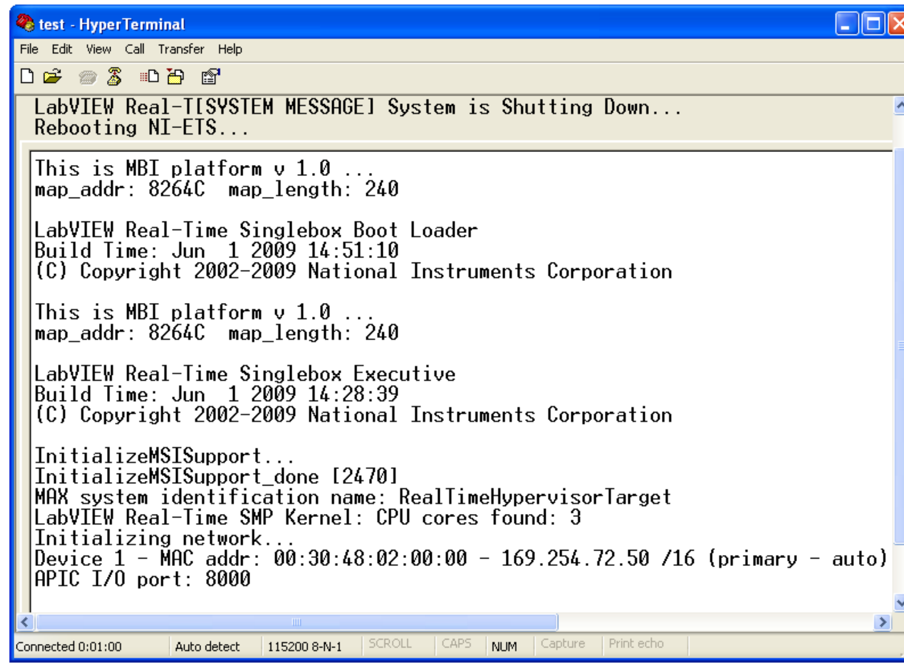
Accessing the Real-Time Target in NI Measurement & Automation Explorer (MAX)



Communicating between OSs

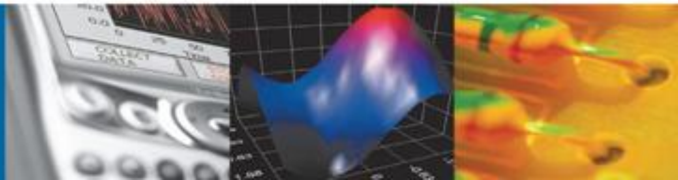


Virtual Ethernet



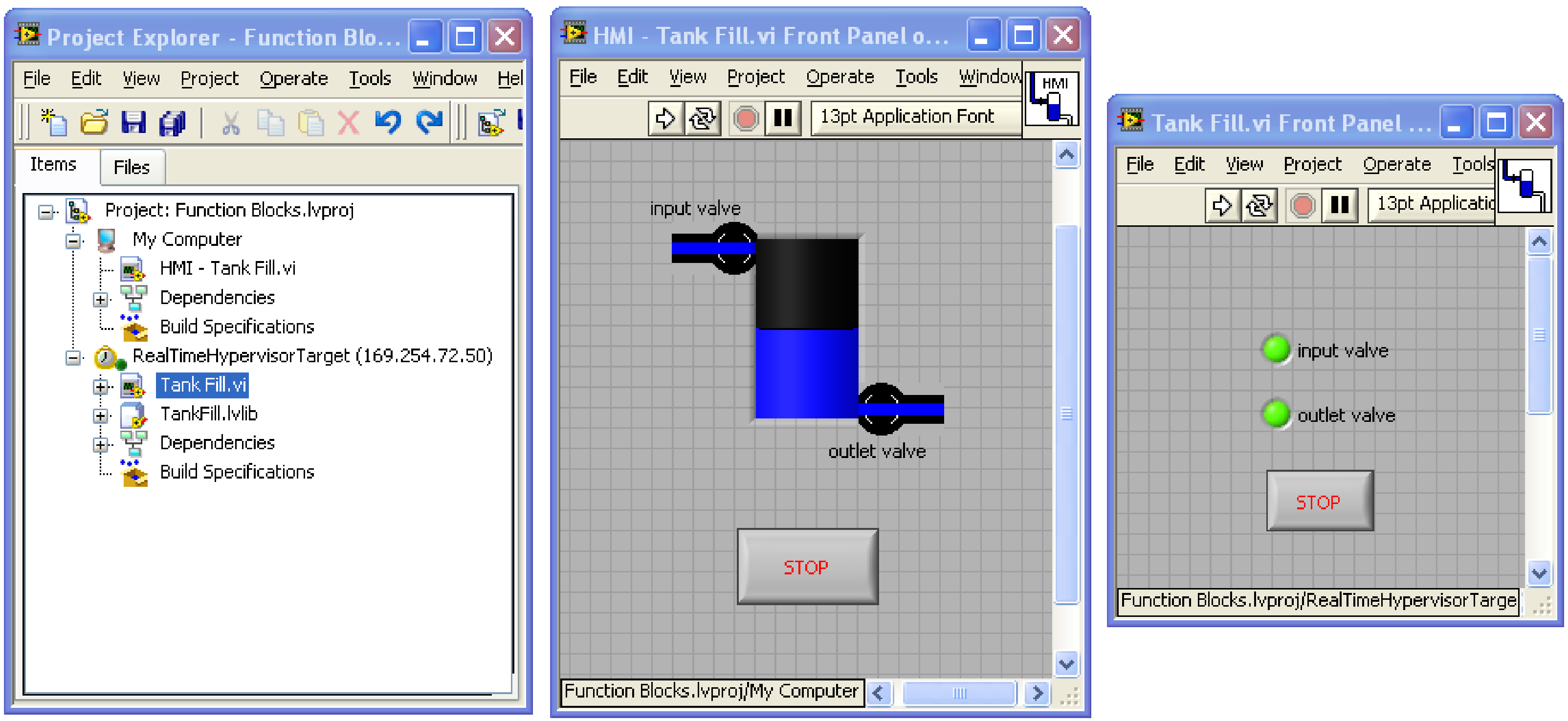
Virtual Console (COM 4)

Demo: Exploring Real-Time Hypervisor Features



LabVIEW Development and Deployment

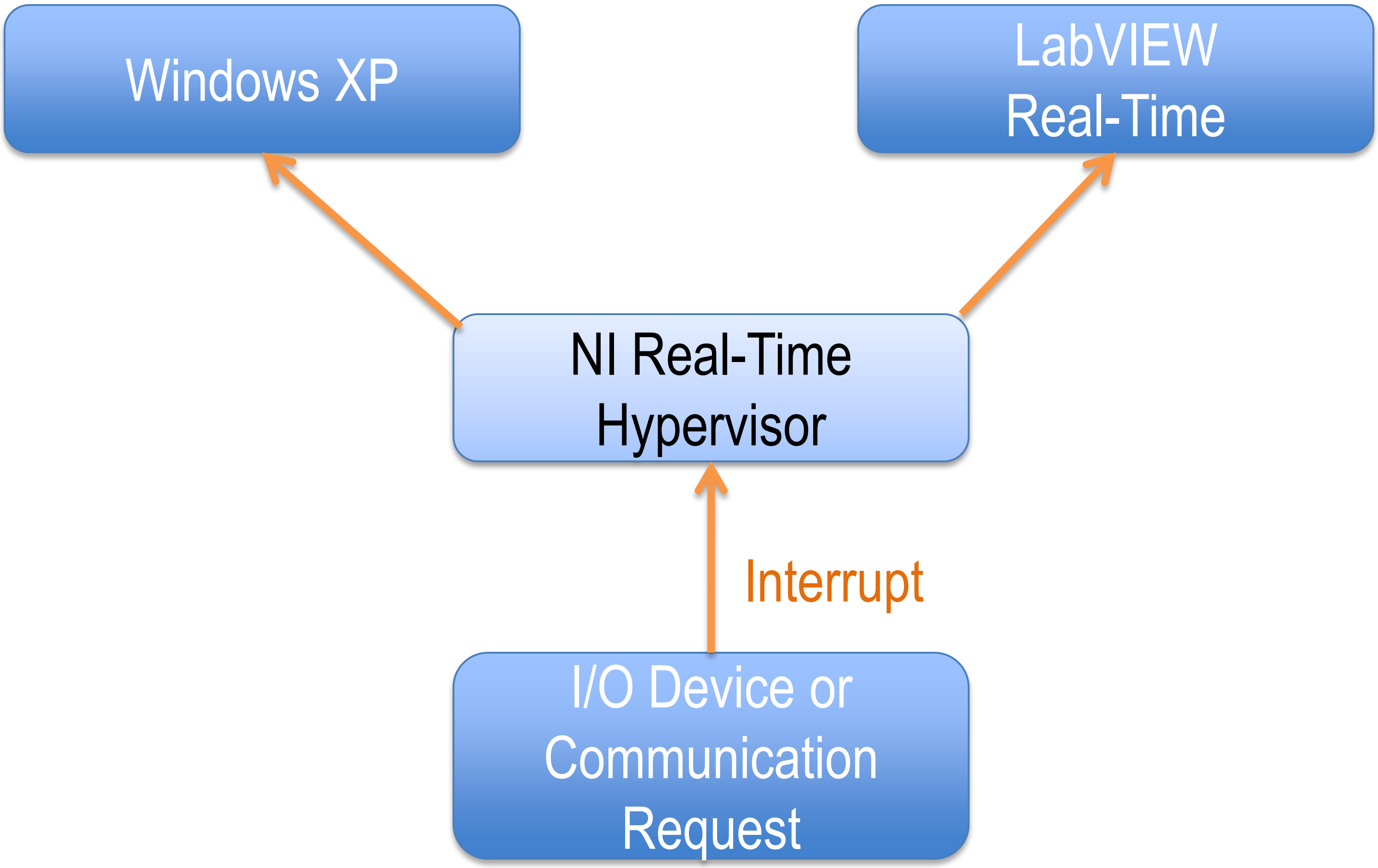
- Extremely similar to traditional NI real-time systems



Demo: Deploying an Example LabVIEW Real-Time Application

Performance and Benchmarks

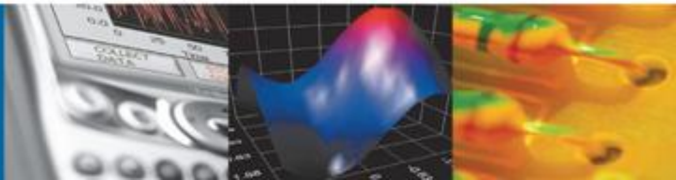
Interrupt Latency and Performance Impact



Benchmarks for Single-Point DAQ Application (Interrupts)

I/O Channels (with PID)	Maximum Loop Rate with Hypervisor (kHz)	Maximum Loop Rate without Hypervisor (kHz)
1	11.5	25.4
4	9.3	22.6
16	7.0	12.4

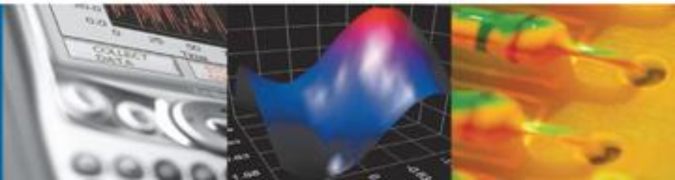
Use polling to improve I/O performance on hypervisor systems



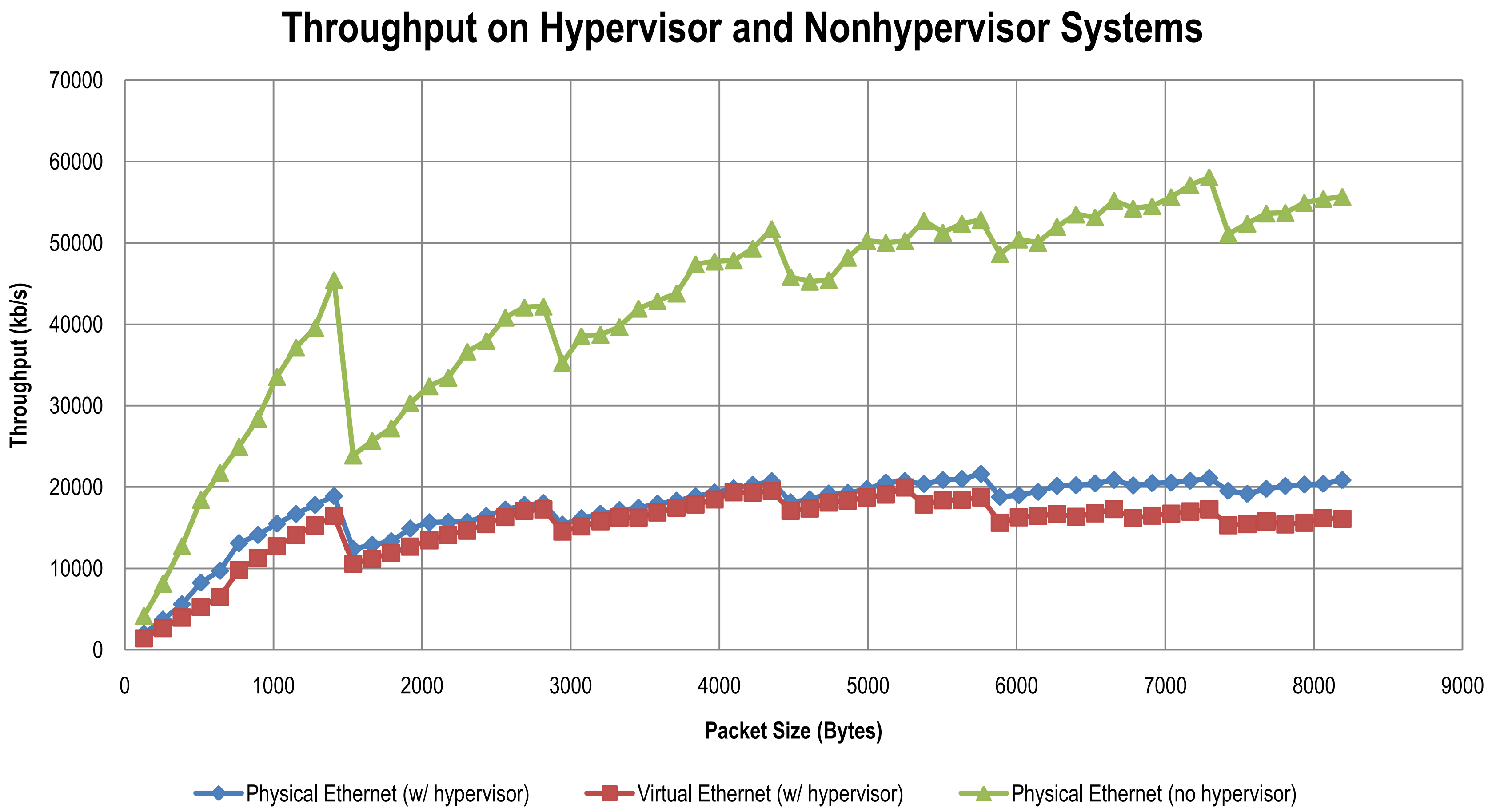
Benchmarks for Typical Large DAQ Application (Polling)

Application	Maximum Loop Rate with Hypervisor (kHz)	Maximum Loop Rate without Hypervisor (kHz)
Large DAQ App.	12.0	14.5

Most LabVIEW Real-Time applications running between 1 and 5 kHz will be able to run at full rate on a Real-Time Hypervisor system



Communication Benchmarks



Case Study

Process Automation: Aircraft Arrestor Test System

Process Automation corp.

- Dynamically testing a system to rapidly decelerate jet aircraft
- Combining real-time simulation, I/O, and user interface on one controller
- Reducing cost and footprint using the Real-Time Hypervisor

"By consolidating the components of our real-time test system onto one controller, the NI Real-Time Hypervisor will reduce our hardware cost and lower our application footprint."
– Greg Sussman, Process Automation

NI Real-Time Hypervisor Ordering Information

- Real-Time Hypervisor and OS software preinstalled
- Supported hardware
  - NI PXI-8108 and PXI-8110
  - NI 3110 industrial controller

- $499 USD (Real-Time Hypervisor Deployment License only)

Additional Resources

- NI virtualization portal (ni.com/virtualization)
  - Background on virtualization technology
  - Real-Time Hypervisor virtual tour
  - Architecture details, benchmarks, and programming recommendations