

Xen in Embedded Systems Ray Kinsella Senior Software Engineer Embedded and Communications Group

Intel Corporation

Legal Notices

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

Intel Virtualization Technology requires a computer system with a processor, chipset, BIOS, virtual machine monitor (VMM) and applications enabled for virtualization technology. Functionality, performance or other virtualization technology benefits will vary depending on hardware and software configurations. Virtualization technology-enabled BIOS and VMM applications are currently in development.

Performance results are based on certain tests measured on specific computer systems. Any difference in system hardware, software or configuration will affect actual performance. For more information go to <u>http://www.intel.com/performance</u>.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order. Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Core Inside, i960, Intel, the Intel Iogo, Intel AppUp, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, the Intel Inside Iogo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. Iogo, Intel StrataFlash, Intel vPro, Intel XScale, InTru, the InTru Iogo, the InTru Inside Iogo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX, Moblin, Pentium, Pentium Inside, skoool, the skoool Iogo, Sound Mark, The Creators Project, The Journey Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2010, Intel Corporation. All rights reserved.



Summary & Agenda

This presentation examines the integration of *Xen* Virtualisation* into embedded systems. It covers effective partitioning of system resources for deterministic embedded applications.

Agenda

- Overview of Xen Virtualisation
 - Types of hypervisor
 - Types of guest
 - Embedded use cases
- Xen in Embedded Systems
 - Partitioning CPU Time
 - Partitioning System Memory
 - Partitioning System I/O
 - Power usage



What is Xen?



- Xen is a bare metal hypervisor
 - Originally designed for data centre or server environments consolidation.
 - A privileged guest Domain 0 (Dom 0) will always exist.
 - Dom 0 owns all devices (PCI, Storage, USB, etc.) and arbitrates their usage between guests.
 - Xen includes mechanisms for device multiplexing such as "split drivers".
- Xen's design goal is the "separation of policy and mechanism".
 - Policy is implemented in Dom 0 Daemons (Xen Daemon).
 - Mechanism is implemented in the Xen hypervisor layer.



Types of guests



Xen can support many diverse guests concurrently

Para-Virtualised (PV) Guests

- Are *"aware"* they are virtualised.
- PV guest will delegate many aspects of operating system function to the hypervisor via kernel hooks aka hypercalls.

Hardware-Virtual-Machine (HVM) Guests

- Are "not aware" that they are virtualised.
- Qemu* emulates common hardware for HVM guests and the operating system loads real world drivers.
- Graphics interface is exported through VNC.



Embedded use cases



Application integration

Integrating new and legacy applications onto same hardware.



Resource isolation

Restricting the resources assigned to each application.



Un-trusted application

Integrate an un-trusted application (a 3rd party application) onto same hardware.



High availability

Ensuring that an application is always available. Standby instances are always ready to run in case of failure.



Xen in Embedded Systems



Virtual CPU Architecture



Virtual CPU (VCPU)

- Are an abstraction layer created by Xen's scheduler.
- Isolates guest from the actual number of physical CPUs.
- Xen has a scheduler similar to an OS scheduler that arbitrates between the guests contending for CPU time based on priority.



Virtual CPU Performance



SSL Encrypt is a simple CPU-intensive application.

- It uses the pthreads library to parallelise an encrypted workload over multiple cores.
- It uses the OpenSSL* libraries to encrypt pools of 64-byte buffers using AES 128-bit CBC encryption.

Equivalent performance native to virtualised



The Credit Scheduler



The Xen *Credit* scheduler

- Default scheduler in Xen 4.0.0
 - The Credit scheduler is proportional fair share scheduler.
 - The Credit scheduler is a work-conserving scheduler.
- Scheduler parameters
 - Cap : Assign a time cap in hundredths of seconds to a guest, (0 N *100), where N is the number of cores in the system.
 - Weight: Assign a weight to each guest (1 65536); default is a weight of 256.

Scheduler partitioning of core time is effective



The Credit Scheduler



The graph above shows the standard deviation of throughput with different scheduler settings, assignment, pinning, and capping.

Modest cost in determinism with scheduler partitioning



Virtual CPU Configuration

Caveat Emptor

Partitioning CPU Time:

• If possible, partition CPU resources by assigning cores to guests rather than using scheduler weighting or caps.

CPU Pinning:

- Make the number of Virtual CPUs equal to the number of cores.
- Use vcpu-pin to assign a guest VCPU to a specific core.

Remember:

- Dom 0 must service the hardware assigned to it.
- In all guests:
 - Switch off unnecessary OS kernel features; i.e., use a tickless kernel...
 - Turn off unnecessary OS services ... e.g., CentOS* Bluetooth* manager.
 - Remove unnecessary drivers; e.g., the USB driver.



Real-Time on Xen



Recommended reading:

<u>Supporting Soft Real-Time Tasks in the Xen Hypervisor</u>; Min Lee, A.S Krishnakumar, P. Krishnan, Navjot Singh, Shalini Yajnik; Georgia Institute of Technology, and Avaya Labs

<u>Extending Virtualization to Communications and Embedded Applications</u>; Edwin Verplanke, Don Banks; Intel Corporation, and Cisco Systems, Inc.; Intel Developer Forum (IDF) 2010

The Chinese Dragon Festival Photo © <u>Walter Baxter</u> and licensed for <u>reuse</u> under a <u>Creative Commons License</u>.



Partitioning System Memory

Bubble Memory + NUMA



- The bubble memory driver can arbitrate (share) memory between guests. This is not the *default behaviour* on Xen.
 - Page allocation incurs a ~16% performance penalty during memory bubbling.
- Mitigate by statically assigning memory to guests. This is the default behaviour on Xen.
 - maxmem: set maximum amount of memory a domain can be allocated
 - memory: set initial amount of memory a domain is allocated

NUMA

- Xen is NUMA-aware
 - Non-Uniform Memory Architecture (NUMA)
 - Switched on in Xen 4.0.0 by default
- Guests are not NUMA-aware (yet!)
 - A patchset has been submitted to Xen, to enable NUMA awareness for PV and HVM guests.





NUMA Optimal Configuration



Guest A - 16 Threads

Native - 16 Threads

Guest A+B - 8+8 Threads

Sub-optimal Configuration

Guest crosses sockets, guest is bound to cores on more than one socket



Optimal Configuration

Guest does not cross sockets, bind guests to cores on one socket only.



Xen Networking – Software Switch



Xen shared device model

Xen software switching mechanism

- Device sharing via the split driver model.
 - The backend driver located in the DomO is responsible for multiplexing guest access to physical hardware.
 - The frontend driver delivers data to the guest, implementing OS device driver interfaces.
- Xen networking
 - 1. Packets are received by the DomO Ethernet* driver .
 - 2. Packets then passed into the OS bridge driver, the destination Ethernet device is looked up, and the packets are forwarded to the Ethernet device driver.
 - 3. Packets are received the Netback driver in DomO and pushed onto a Xenbus ring.
 - 4. Packets are popped off the Xenbus ring by *Netfront* driver and passed to the guest operating system.



Xen Networking – Hardware Switch



Xen device passthrough model



SR-IOV hardware switching

Software Switch VMDq SR-IOV SW SW HW SW HW RX ТΧ RX ТΧ RX ТΧ MAC Filter IOMMU **RX: Receive Queue TX: Transmission Queue** HW: Hardware Queue MAC Filter SW: Software Queue



• Single Root I/O Virtualisation (SR-IOV)

Built on the following technologies available \mbox{Intel}^{\otimes} VT-c enabled Network Interface Cards (NIC)

- I/O Acceleration: Intel[®] VT-d (IOMMU)
- Filtering Technology: Virtual Machine Device Queues (VMDq) -Hardware MAC Filtering
- Queuing Technology: Multiple RX + TX Hardware Queues

Each guest has an exclusive NIC that has been virtualised in hardware (SR-IOV).

- Xen Passthrough
 - PCI configuration space is still owned by DomO, guest PCI configuration read and writes are trapped and fixed by Xen PCI passthrough

Partitioning System I/O

Hardware vs. Software Packet Switching



35x performance increase on small packet sizes

- Hardware
 - 2 x Intel[®] Xeon[®] Processor E5645 (12M Cache, 2.40 GHz, 5.86 GT/s Intel[®] QuickPath Interconnect) 80W Thermal Design Power (TDP)
 - Intel[®] 82599 10Gb Ethernet Controller with SR-IOV capabilities
- Software
 - Measured with Xen 4.0.0 with CentOS 5.4 Guests
 - 1 Socket/6 Cores used to route traffic



Power Usage



Equivalent performance native to virtualised

- Hardware
 - 2 x Intel[®] Xeon[®] Processor L5530 (8M Cache, 2.40 GHz, 5.86 GT/s Intel[®] QPI) 60W TDP
 - Intel[®] Server Board S5520HC
 - Enermax* EVR1050EWT* Power Supply Unit 88.61% efficiency
- Software
 - Measured with Xen 4.0.0 with CentOS 5.4 Guests and CentOS 5.4 Native



Conclusions

- Equivalent performance native to virtualised.
- Potential benefits of virtualisation for embedded system designers:
 - Greater system security.
 - Greater system determinism.
 - Improved resource isolation.
 - Controlled 3rd party platform access.



Intel®