Mini-VM – Extending KVM Toward Embedded Systems

Jun Nakajima

Intel Open Source Technology Center

KVM Forum 2008

Agenda

- Atom Processor
- What's Mini-VM and Why?
- Prototype and Current Status
- Next Steps

Legal Disclaimer

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. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.

Intel may make changes to specifications and product descriptions at any time, without notice.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel, processors, chipsets, and desktop boards 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.

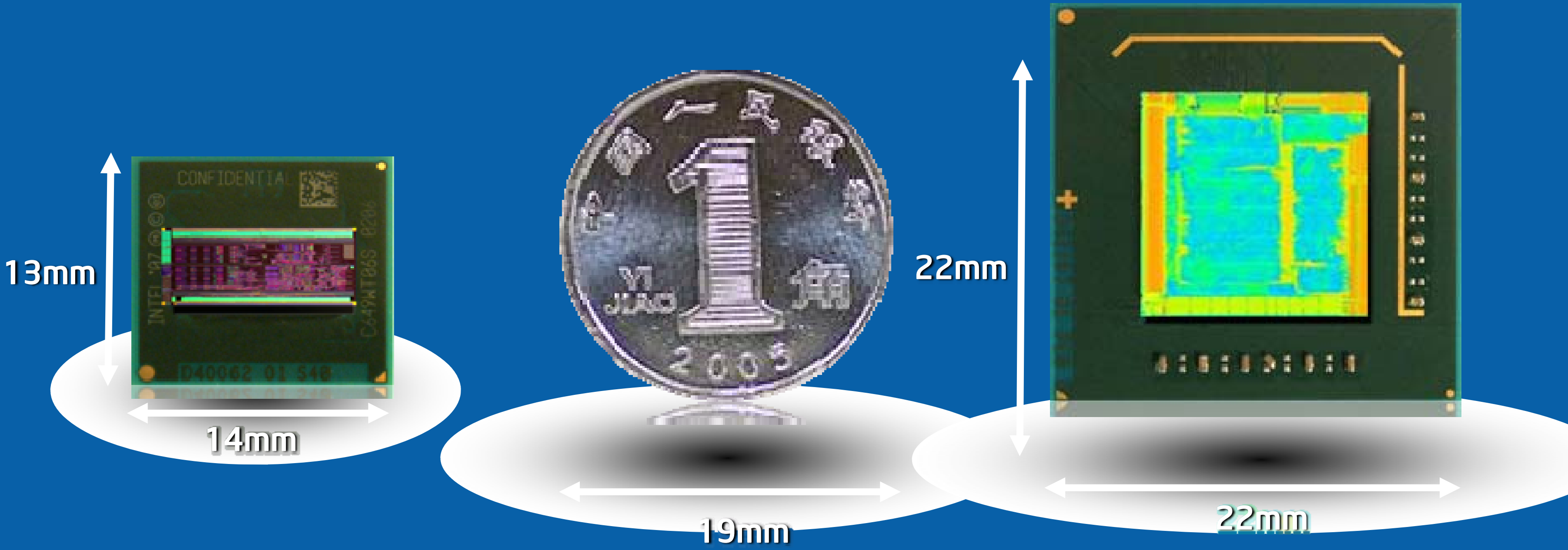
Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

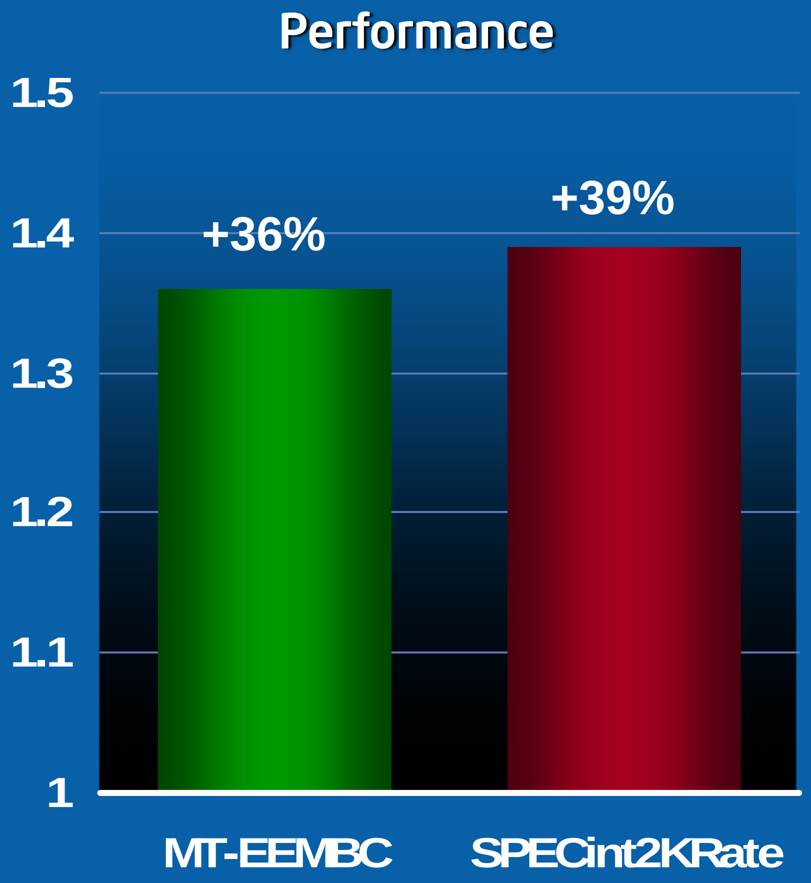
Throughout this presentation:
VT-x *refers to Intel® VT for IA-32 and Intel® 64*
VT-i *refers to the Intel® VT for IA-64, and*
VT-d *refers to Intel® VT for Directed I/O*

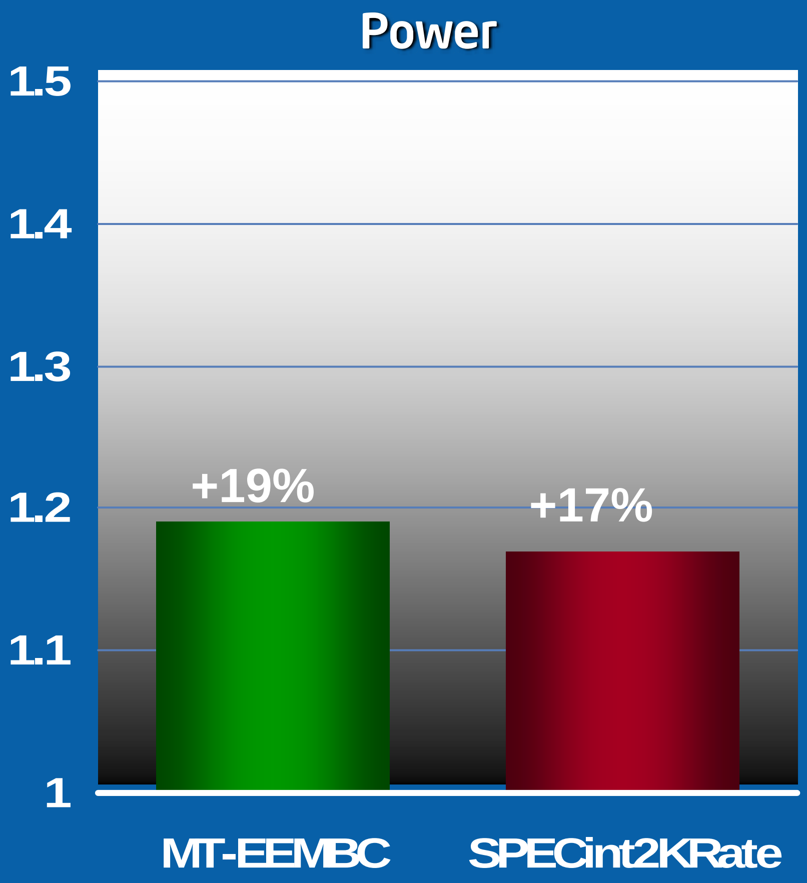
Intel® Centrino® Atom™ Processor Technology

(Formerly Codenamed Menlow)



Performance / Power Efficiencies





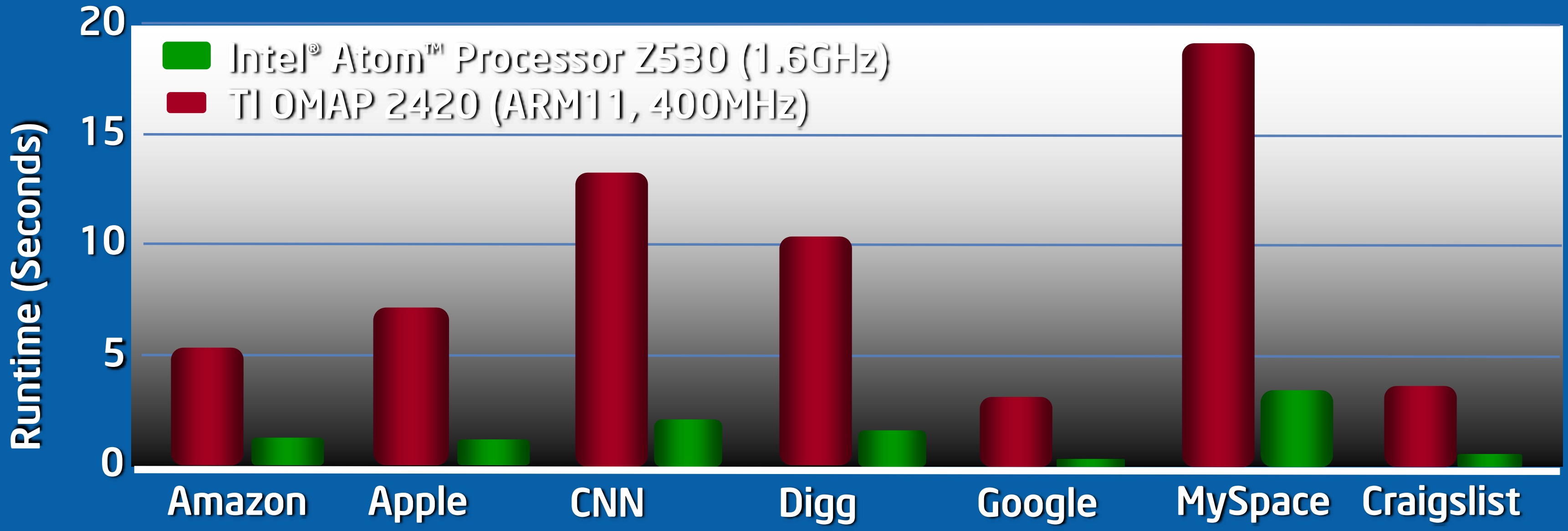
Hyperthreading Performance Increase In An In Order Machine*

Source* : Intel Testing. Specint2k and EEMBC run in Single Threaded / Hyper Threaded Mode on Linux. For Performance the score for each binary is calculated based on the runtimes; For Power, the effective capacitance or C-dyn is measured per binary on each of the benchmark while running in ST and HT modes. The difference in C-dyn and thus total power difference is calculated for ST and HT modes. Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

Webpage Render Performance

Performance Comparison on Webpage Render*

(Browsing Over SSD – Network Dependencies Removed)



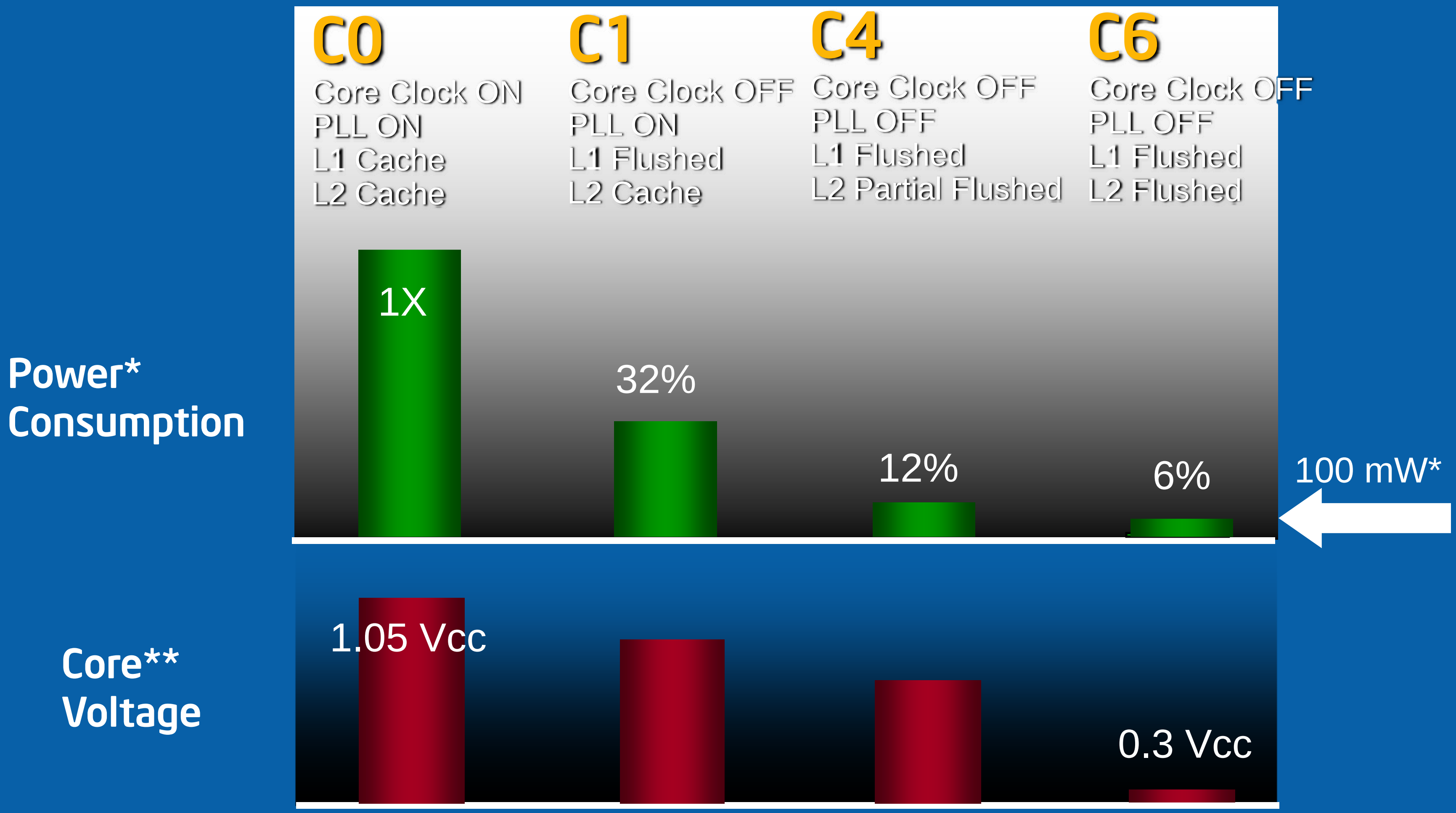
4.1-6.5X Competitive Advantage

•Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

•TI OMAP2420 (ARM11, 400 MHz) results are based on a standard Nokia N810 device purchased in retail

Power Consumption

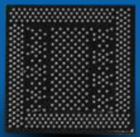
*New Thresholds: Intel Deep Power Down C6**



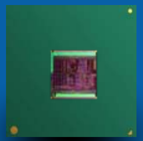
* Absolute Power numbers shows results of average power measurements of MM05 benchark. Results for each state will vary based on the type of part and SKU. Example shown is measured data from 1.86G Silverthorne Part – all states at 50C. ** Core Voltage shown is measured data from 1.86G Silverthorne Part. Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

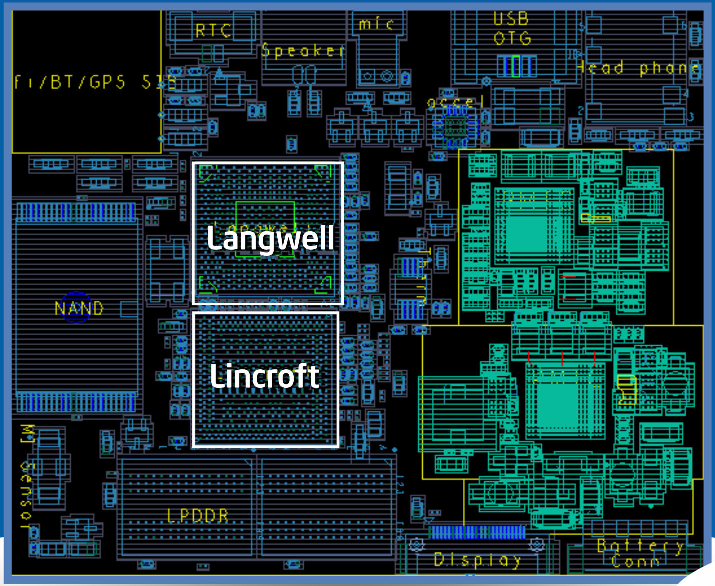
Moorestown

Lincroft



Langwell





More at Fall IDF …

All plans, features and dates are preliminary and subject to change without notice.
*Other brand and names are the property of their respective owners.

PC vs. Devices

- VM in KVM (along with Qemu) means “PC”
  - Legacy devices, interrupt controllers, timers, ACPI/BIOS, PCI devices, monitor, keyboard, mouse, etc.
- There are various devices or computers that are not compatible with PC
  - Network routers, …, robots, …, toasters, …, PDAs/MIDs, …
  - Some can afford very small amount of memory (e.g. 128MB)
- And various operating systems and apps have been developed for those
  - Porting such (legacy) OS, drivers, and apps to “PC” is not straightforward

Benefits of Using Virtualization for Embedded Systems

- Portability & Maintainability
  - Provides simplified and uniformed VM to minimizing porting and maintenance efforts
  - Once virtualized, it's independent of H/W
- Scalability & Consolidation
  - Legacy operating systems often support UP only
  - Multiple instances of VMs
- Reliability & Protection
  - Tolerate and isolate fatal errors in legacy OS guest and software to avoid system crash
  - Sandboxing

What's Mini-VM and Why?

- Bare minimum and simple VM
  - CPU(s), memory, abstracted (PV) devices
    - Timer, front-end devices (or virtio)
  - Start from protected (or 64-bit) mode with paging enabled; no real mode; No BIOS
- Protected execution environment by H/W
  - Run under H/W-assisted virtualization
  - Allow Ring-0 operations, eliminating burden of para-virtualizing CPU
- Low virtualization overheads
  - Use hybrid virtualization (PV + H/W-assisted virtualization)
  - Real-time (e.g. direct paging mode)

Implementation

- Mini VM sounds like Mini-OS in Xen
  - Mini-OS uses Xen API for PV
    - VCPU, timer, event channels, front-ends, SMP, etc.
  - However Mini-OS is pure PV VM; VCPU is different from usual x86
    - No privileged instructions, kernel runs in ring 1 or 3.
  - Mini-OS does not run in H/W virtualization container
- Solution:
  - Run Mini-OS in KVM removing PV API form Xen VCPU
    - Restore IDT, GDT, privileged instructions, etc.
  - Use xenner
    - Change it to run ring 0 PV guest

Prototypes and Current Status

Started running with (modified) xenner

3 patches (by Disheng Su, disheng.su@intel.com)

1. Mini-OS (64-bit, 32MB)
   - Replace VCPU PV ops with x86 instructions
   - Mostly in
     - `extras/mini-os/arch/x86/traps.c`
     - `extras/mini-os/arch/x86/x86_64.S`
2. Xenner
   - Hypercall handling for H/W-based VM
   - Allow Ring 0 guest
     - `hypercalls-dead.c`
3. KVM
   - Direct-page support and misc

Next Steps

- Check Real-time characteristics
  - E.g. Shadow page table vs. direct paging mode
- More contents/examples of mini-VMs
  - Assign particular H/W device(s)
- Decide code structure
  - Merge into xenner or sperate tree
    - Xenner can be part of Qemu ; too big for Mini-VM

intel®
Leap ahead™

Prototype (1/3)

- Patches have been developed by Disheng Su (desheng.su@intel.com)
- Patch to mini-OS in Xen
  - – Replace VCPU PV ops with x86 instructions

```
extras/mini-os/arch/x86/setup.c      |    5
extras/mini-os/arch/x86/traps.c      |  261 ++++++++++++++++++++++++++++++++---
extras/mini-os/arch/x86/x86_64.S     |  219 +++++++++++++++++++----------
extras/mini-os/console/console.c     |    2
extras/mini-os/events.c              |    5
extras/mini-os/fbfront.c             |    6
extras/mini-os/hypervisor.c          |    4
extras/mini-os/include/x86/arch_mm.h |   13 +
extras/mini-os/include/x86/os.h      |   14 +
extras/mini-os/kernel.c              |   16 +-
extras/mini-os/mm.c                  |    7
extras/mini-os/xenbus/xenbus.c       |    5
xen/include/public/xen.h             |    7
```

Prototype (2/3)

- Patch to xenner
  - Hypercall support for H/W virtualization
  - Allow Ring 0 guest

```
build.c             |   74 +++++---
emu64.h             |    4
evtchn.c            |   25 ++
hypercalls-dead.c   |  435 +++++++++++++++--------------------------------
hypercalls.c        |   43 ++++-
include/linux/kvm.h |    7
kvmbits.c           |   18 +-
libkvm.c            |   18 ++
mm.c                |   10 -
netbackd.c          |    4
xenner.c            |    5
xenner.h            |    6
```

Prototype (3/3)

- Patch to KVM
  - Hypervisor mode for direct paging mode

```
arch/x86/kvm/mmu.c   |   32 ++++++++++++++++++++++-
arch/x86/kvm/vmx.c   |   11 ++++---
arch/x86/kvm/x86.c   |    9 ++++--
include/linux/kvm.h |    7 ++++
virt/kvm/kvm_main.c |   74 ++++++++++++++++++++++++++++++++++++++++++++++++++++++
5 files changed, 123 insertions(+), 10 deletions(-)
```