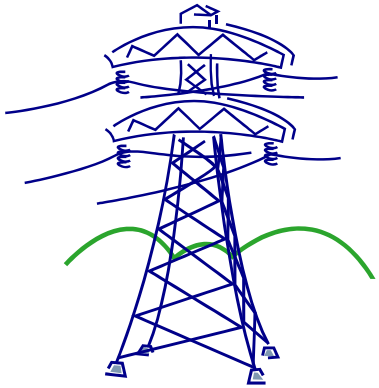
Xen Power Improvements



Will Auld, Yang Z Zhang, Winston Wang

Intel Corporation

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.

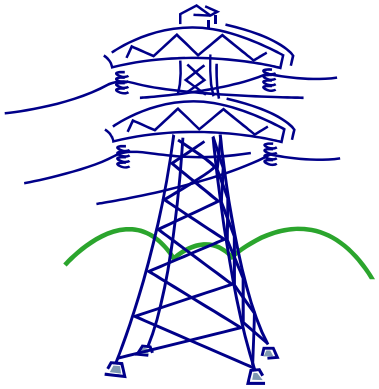
*Other names and brands may be claimed as the property of others.

Agenda

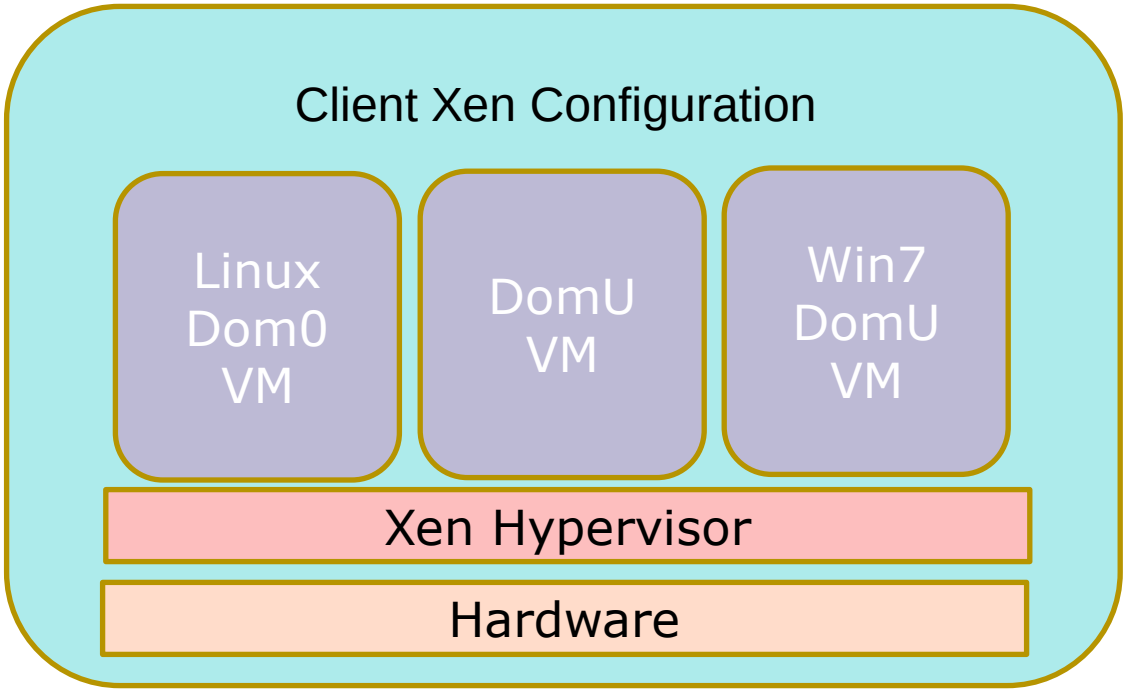
- Background
- Power saving in client
- Power saving in server
- Summary

Room to save POWER

- **Ideal/standard ➔ Native OS power consumption**
- **Reality ➔ Hypervisor power consumption**
- **LARGE DELTA (~40% for client at start)**

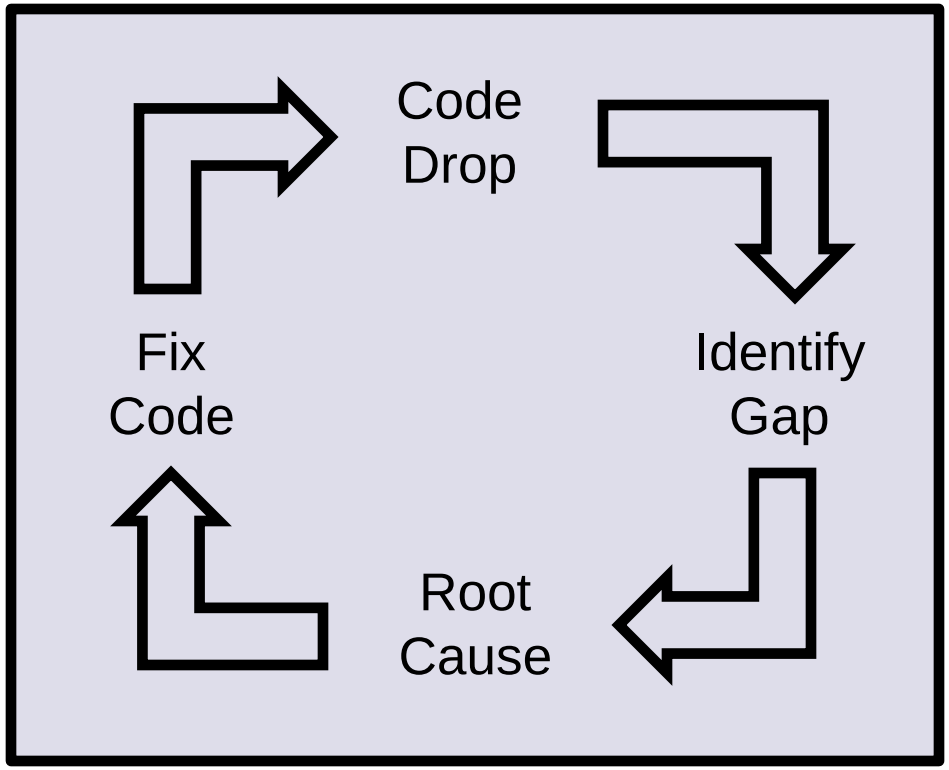


Client architecture



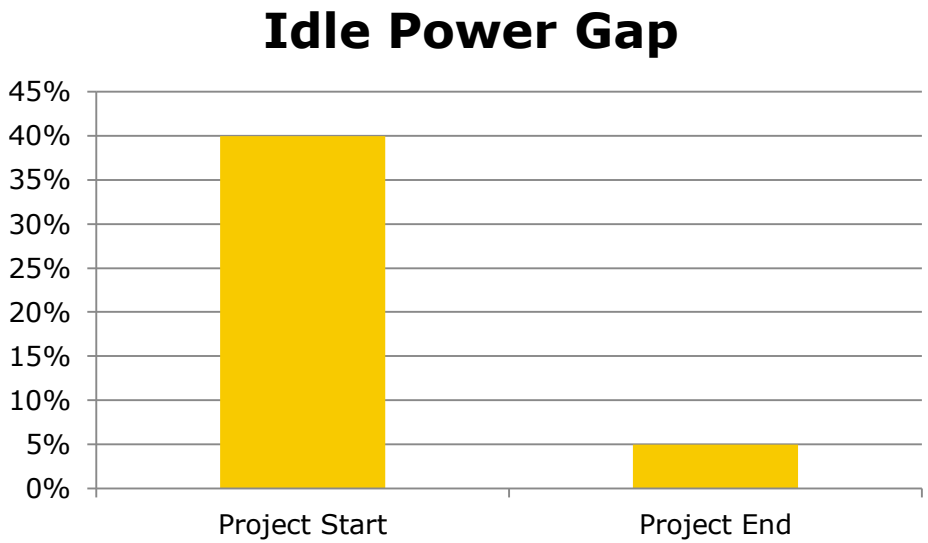
Goal

- Native OS power efficiency
- Close the Power gap with Native Win7



Current results

- ~40% idle power gap 2 years ago
- ~5% idle power gap now



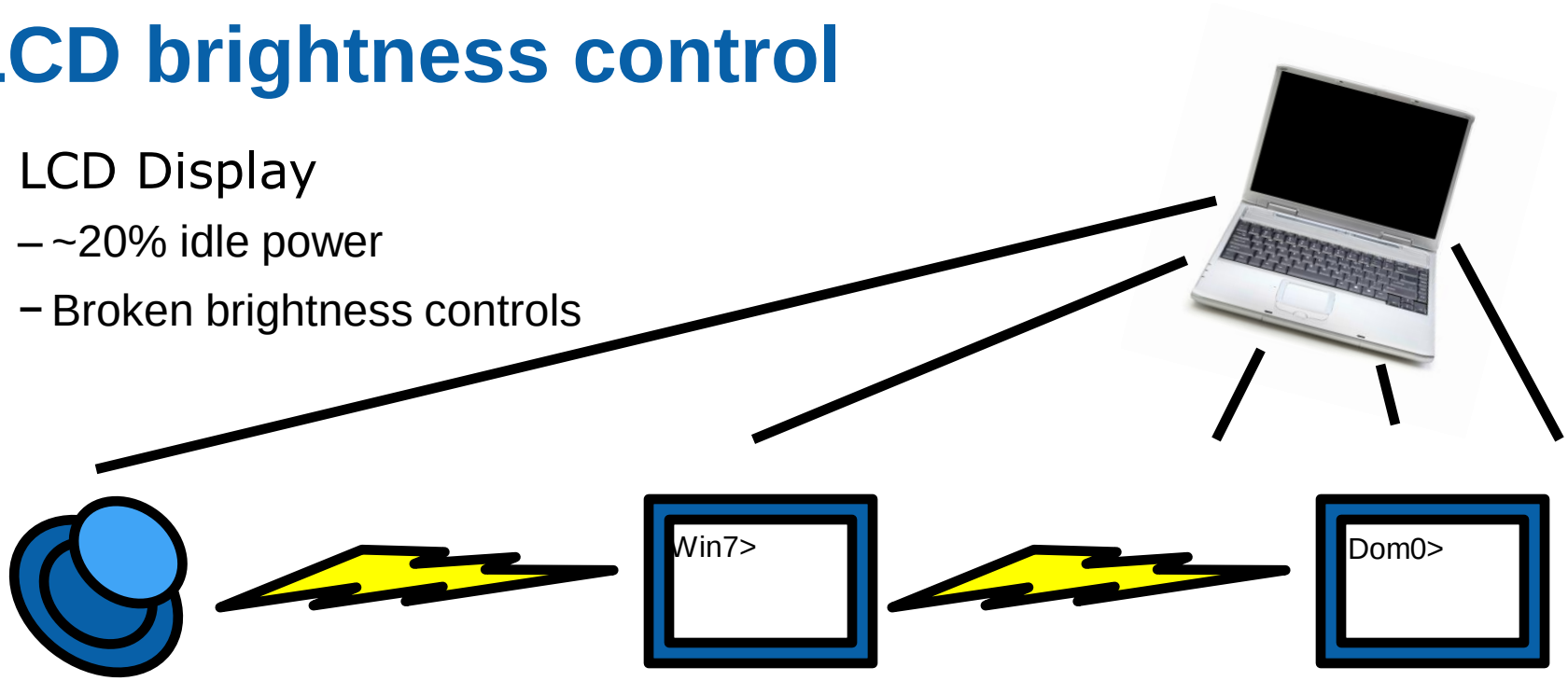
- More?
- Increasingly harder to extract

LCD brightness control

LCD Display

– ~20% idle power

– Broken brightness controls



Fix:

– Added emulation of ACPI video extension

  – Specifically, brightness control methods _BCL, _BCM, and _BQC

  – Added to VM guest ACPI BIOS

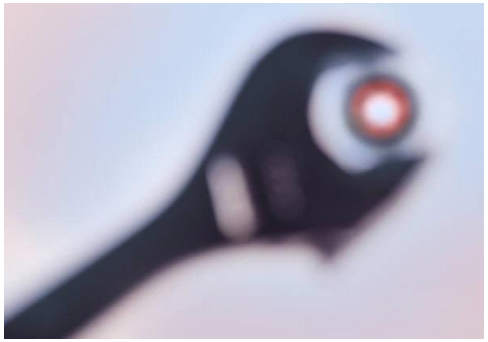
  – Pass through control knob output to Dom0 take platform action

– Make sure Dom0 LCD brightness is really working

Runtime IO power management

Dysfunctional IO power management

- ~15% Idle power
- 1st available in 2.6.32 kernel, but:
  - not functioning correctly



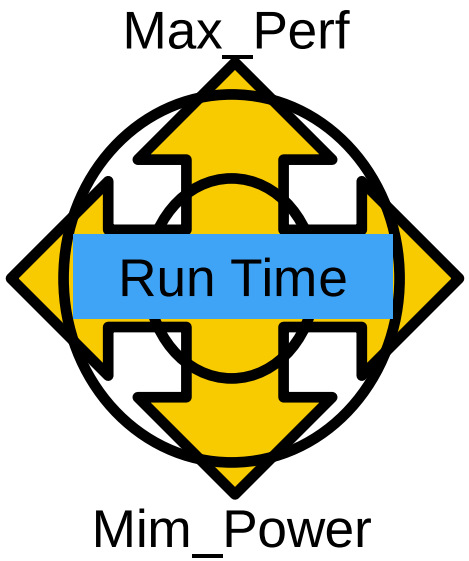
Fix:

- Enable energy-saving states at run time and auto suspended when idle
- Gap dropped from ~25% to 6.8% after fix
  - HP 8440p mobile platform based on Nehalem processor

ATA_link power

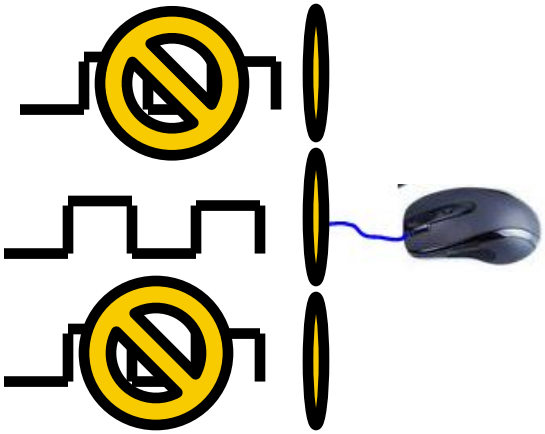
ATA_link static power setting

- ~6% idle power in max_performance
- But performance suffers with min_power
- Even worse:
  - All SCSI hosts active with/without attached devices



Fix:

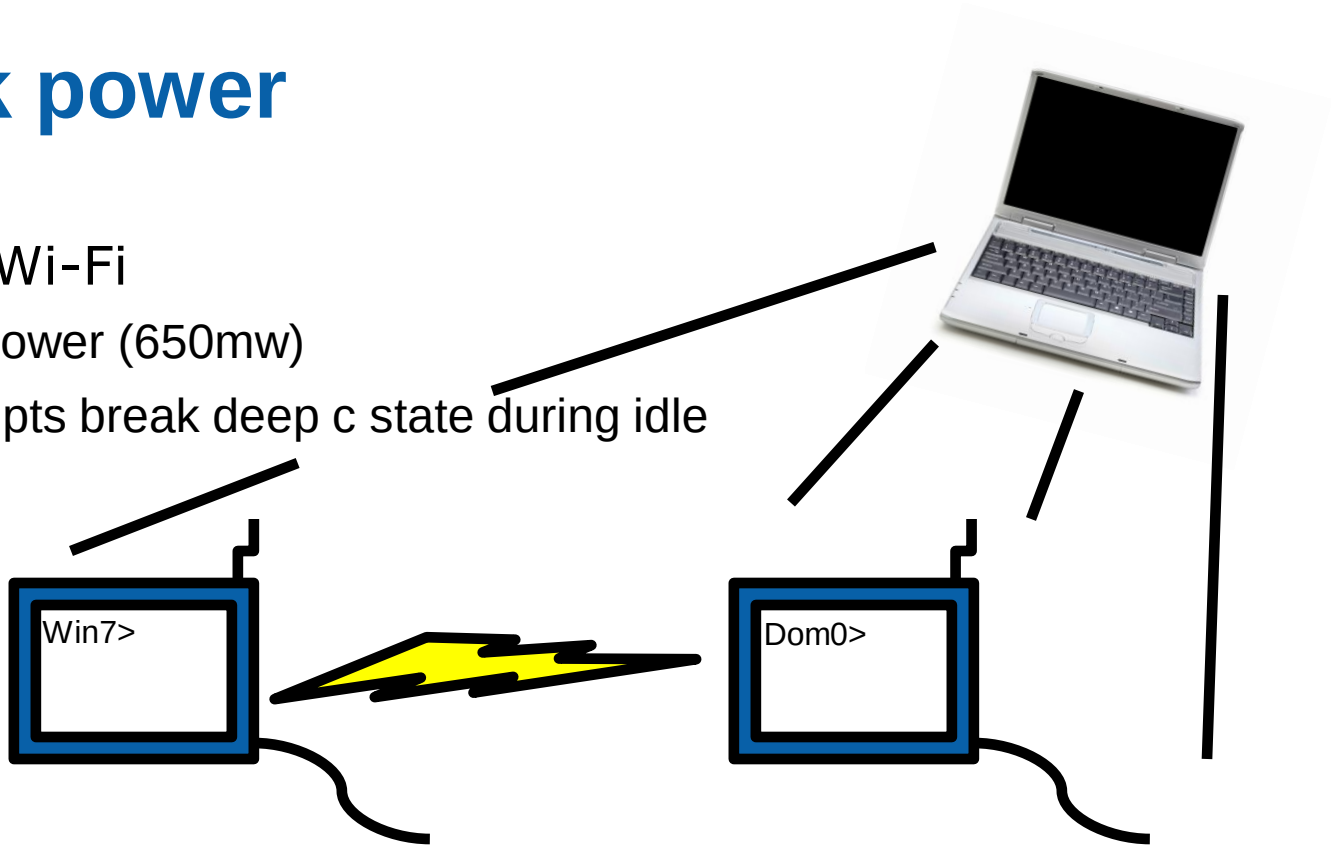
- Runtime update for ATA_link power setting
  - Toggle min_power / max_performance, as needed
- Disable clocks on deviceless ports



Network power

Wired and Wi-Fi

- ~16 % idle power (650mw)
- Many interrupts break deep c state during idle



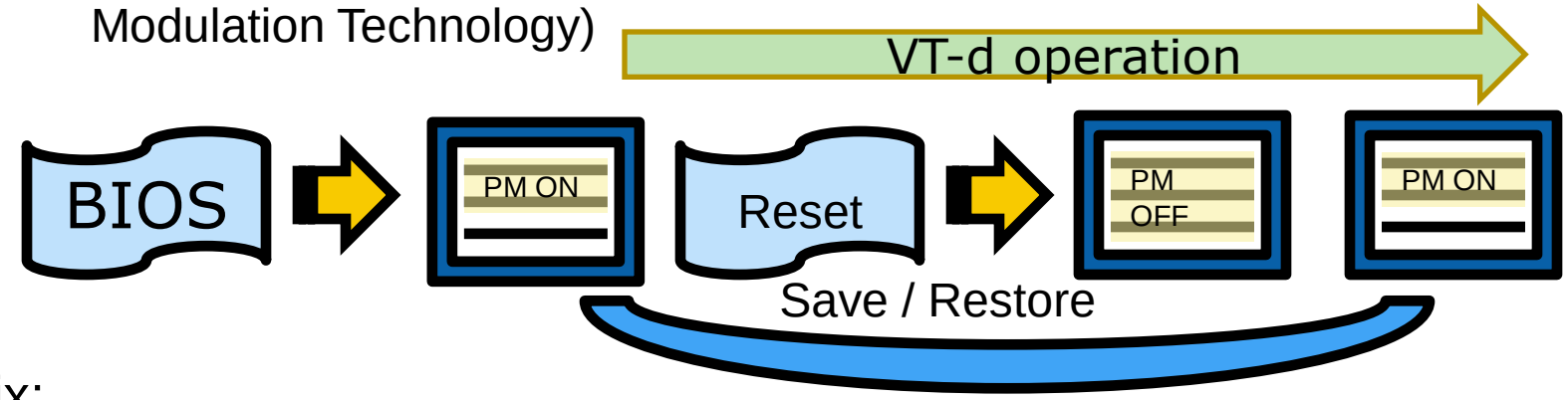
Fix:

- Enable Wi-Fi and E1000 power saving mode in Dom0
- Add Win7 power management PV driver to pass control settings to Dom0

GFX power management

iGFX power management inactive

- ~16% idle power (650mw)
- VT-d requires device reset
  - Reset clears all regs including BIOS enabled power management regs
    - Disables: RC6 (render standby), turbo, and GPMT (Graphics Power Modulation Technology)



Fix:

- Save/Restore PM registers around FLR

Client summary

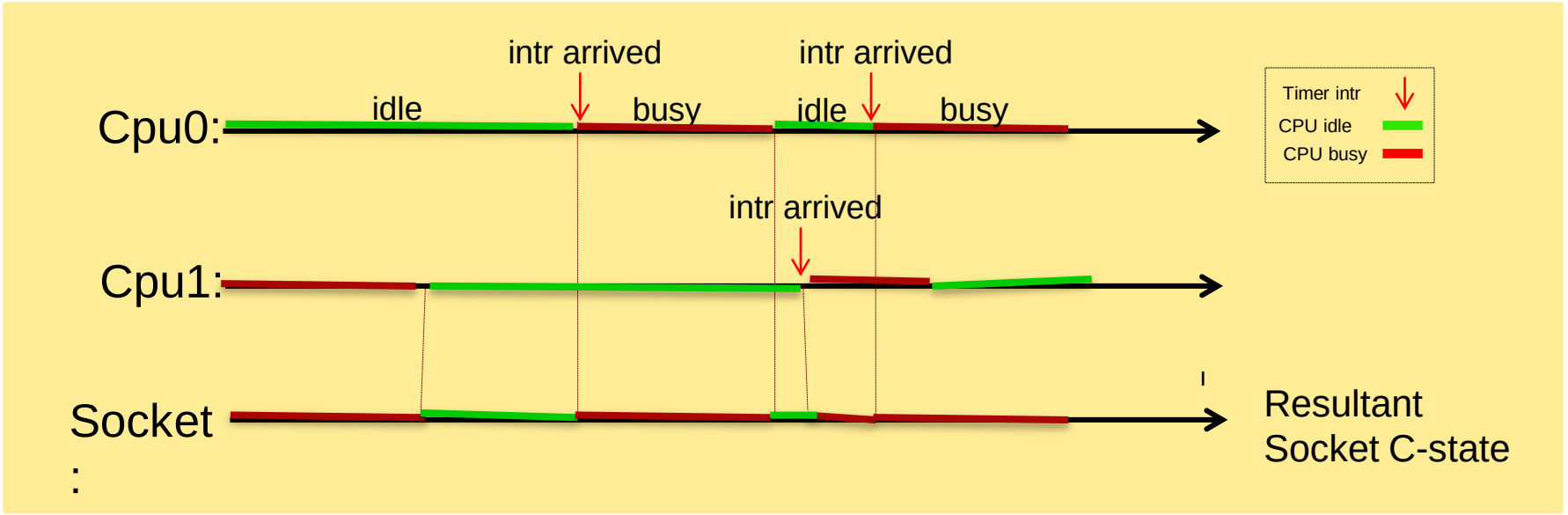
- Started with a ~40% gap
- Ended with ~5% gap
- Greatly improved and got close to the goal

Server power savings -- increasing idle time

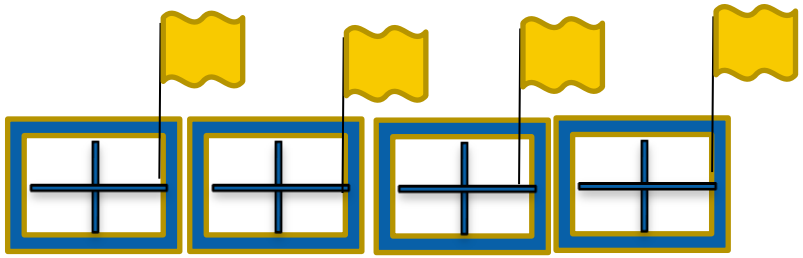
- **Timer alignment**
- **Power aware scheduling**
- **Reducing periodic tasks**

Timer alignment

- Independent, frequent timer interrupts ➔
- Frequent wake-ups
- Reduced idle time, greater power consumption



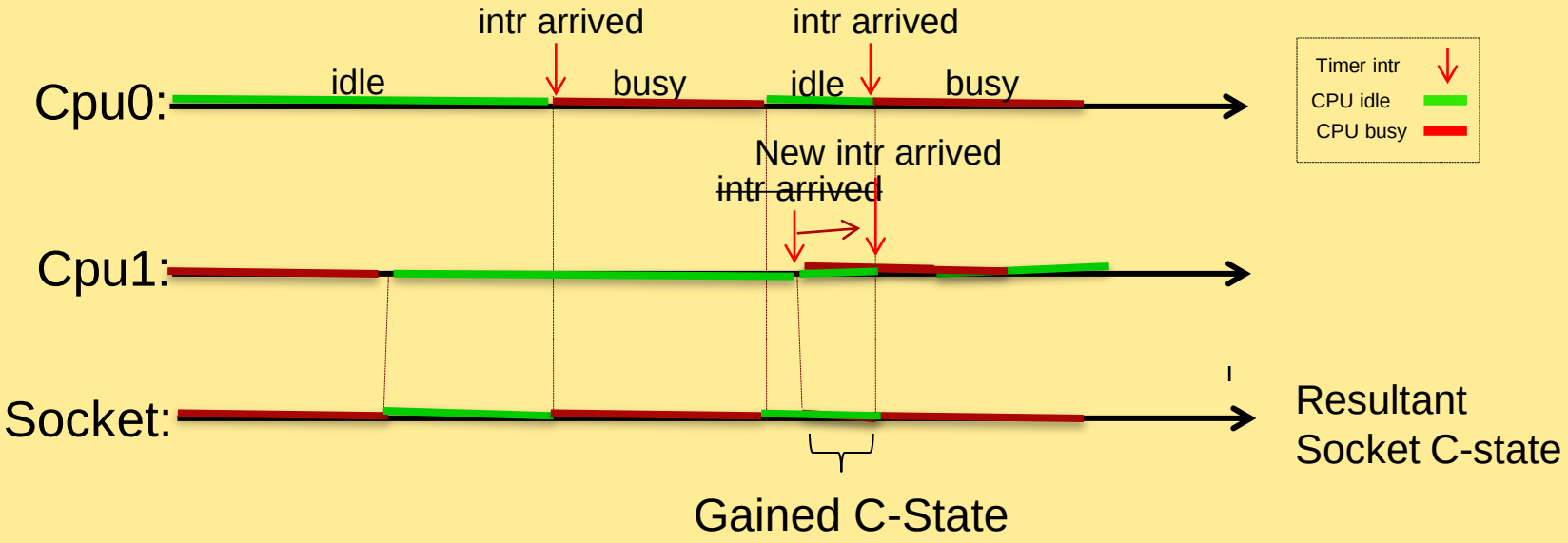
Timer alignment



- Proposal
  - Configurable timer consolidate window, such as 50 ns
  - Compute timer interrupt moment
  - Shift timer handle moment to next timer consolidate moment
- Benefit
  - Fewer interrupts ➔ longer idle time ➔ power savings

- Challenges
  - Guest schedule impact– performance impact
  - Cross CPU timer synchronization
  - IPI frequency and synchronization

Timer alignment



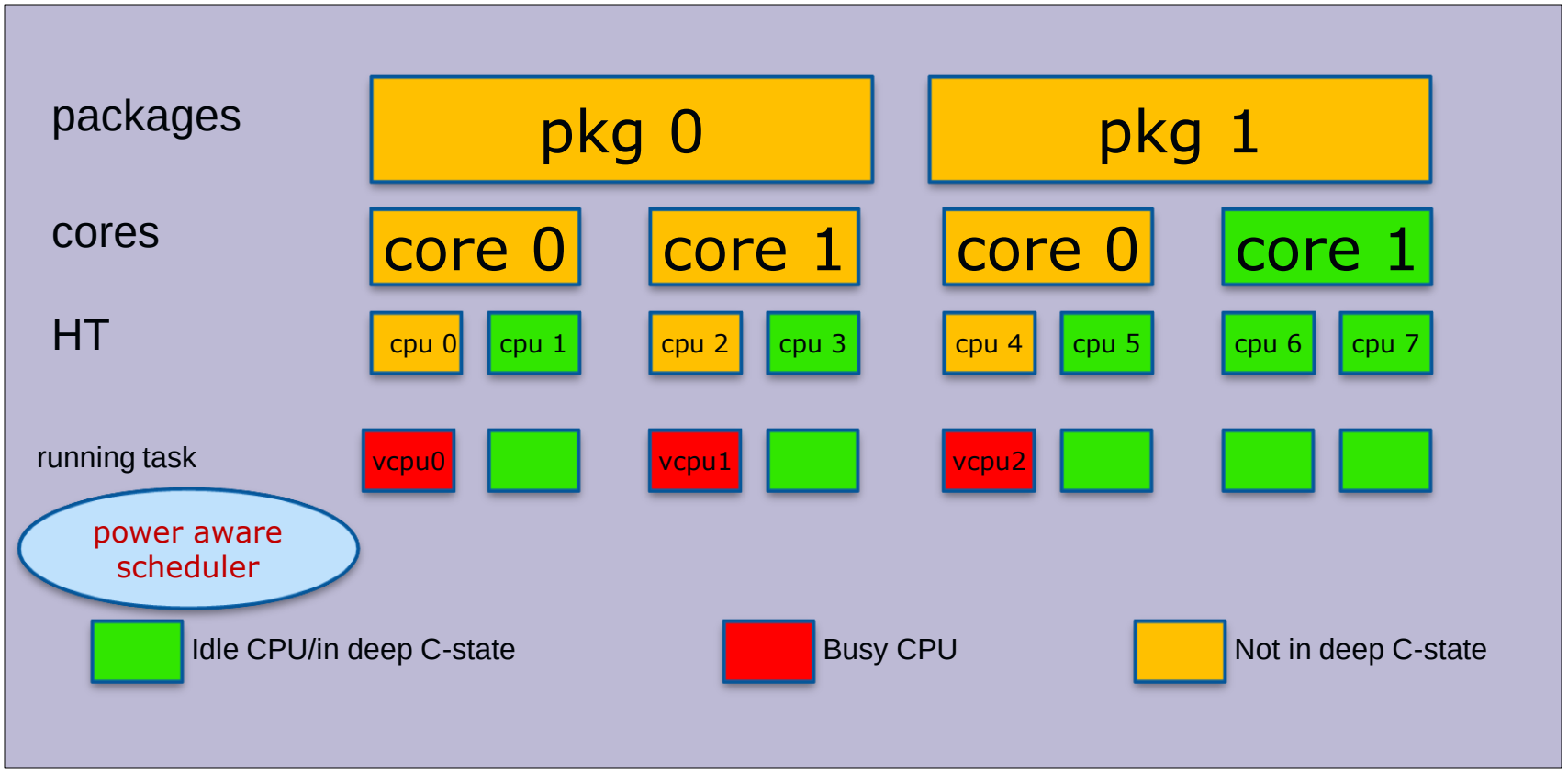
- Shifting CPU1's interrupt to match CPU0's ➔Nice gain in C-State
- Repeated over and over adds up

Power aware scheduling

- ACPI modes –
  - Performance → Power hungry mode
  - Energy mode → Power savings mode
  - Balanced

- Task to Scheduling
  - Performance
    - Schedule vCPUs one per physical core before pairing
  - Energy
    - Schedule vCPUs one per logical core →
      - power down more cores →
      - power down more sockets

Power saving scheduler

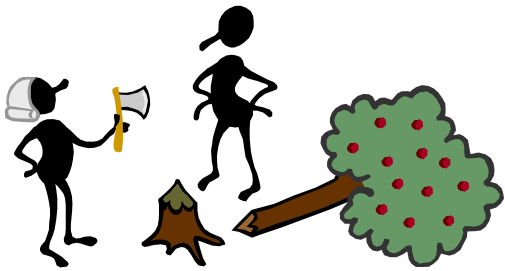


Reduce periodic activity

- Power-unfriendly RTC emulation:
  - VMM updates RTC clock twice per second
  - Solution
    - Update RTC clock only on Read

If a clock ticks where no one can see it, does the time change?

- Frequent Wake-ups to check buffered I/O:
  - Wakeup multiple times a second (Polling model)
  - Solution (Push model)
    - Event channel to notify buffered I/O change status



No more polling

Server summary

- Significant areas of work
- Need to quantify the impacts

Overall summary

- Every component counts – software and hardware
- Make sure the basics are working
- Still more to do

Questions?