The On-going Evolutions of Power Management in Xen

Kevin Tian

Open Source Technology Center (OTC)

Intel Cooperation

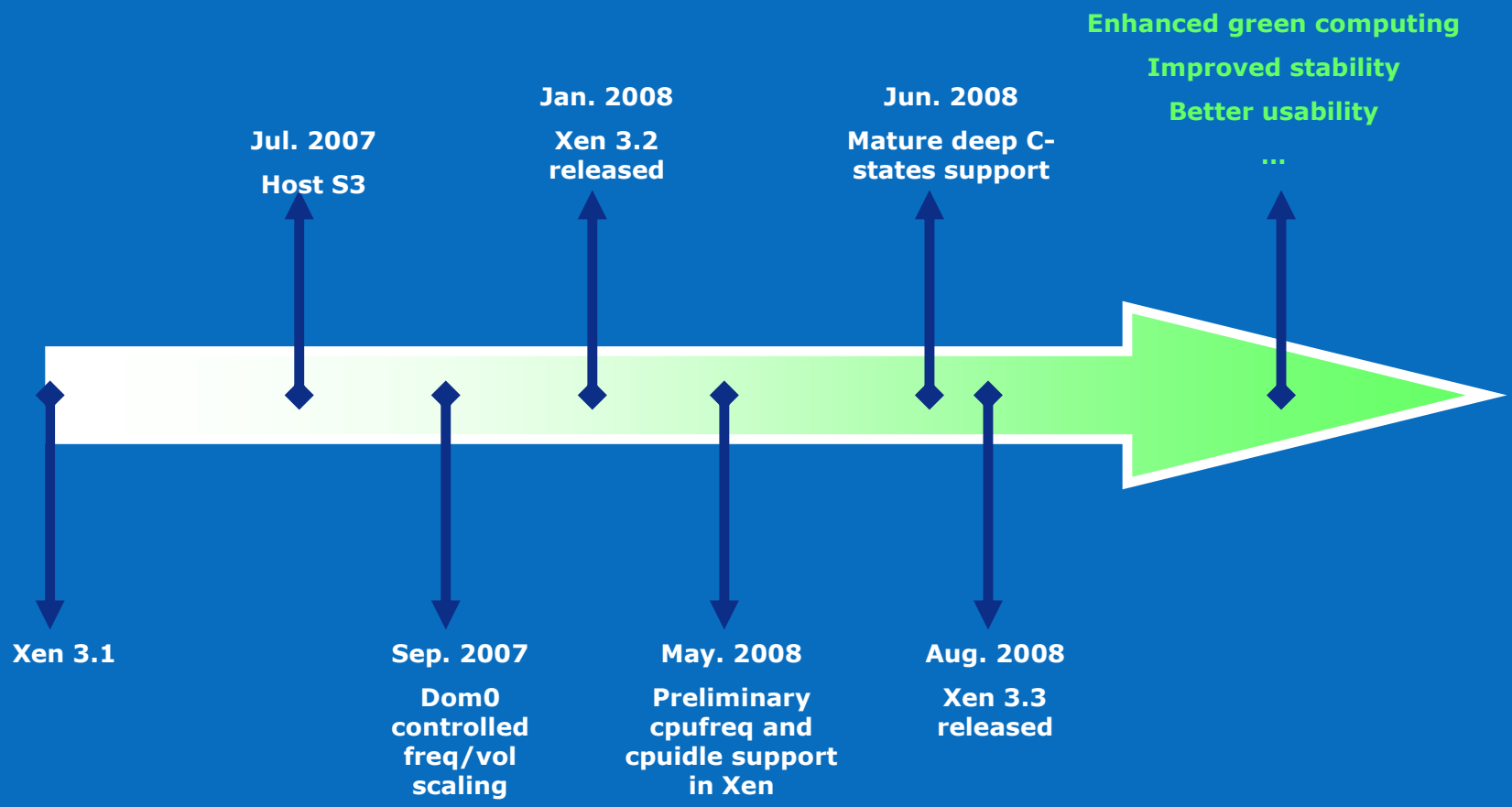
Open Source Technology Center

Software and Solutions Group

Agenda

- Brief history
- Evolutions of Idle power management
- Evolutions of run-time power management
- Tools
- Experimental data about Xen power efficiency
- Power impact from VM

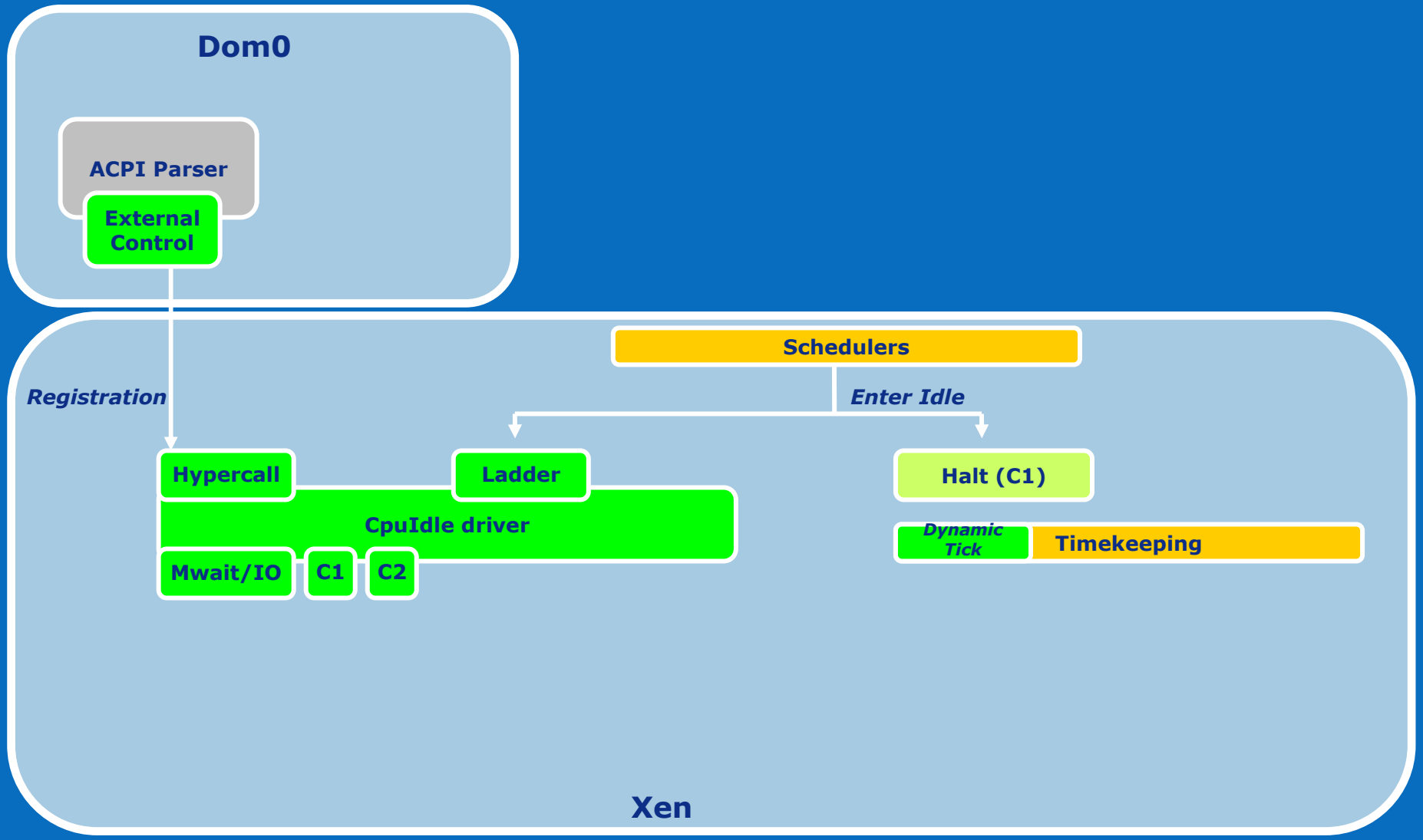
Brief History



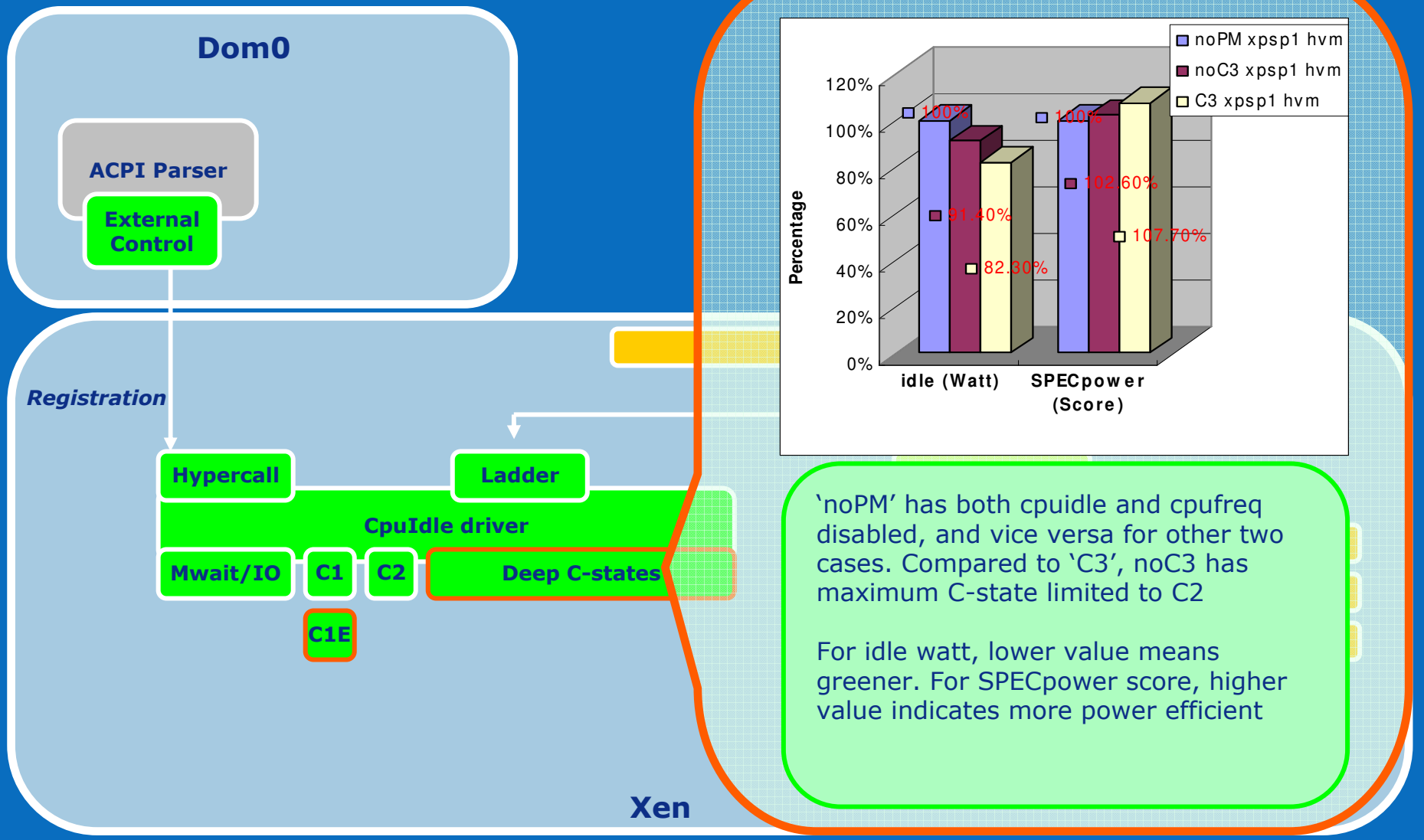
Evolutions of Idle power management

(Xen cpuidle)

Xen Summit Boston 2008



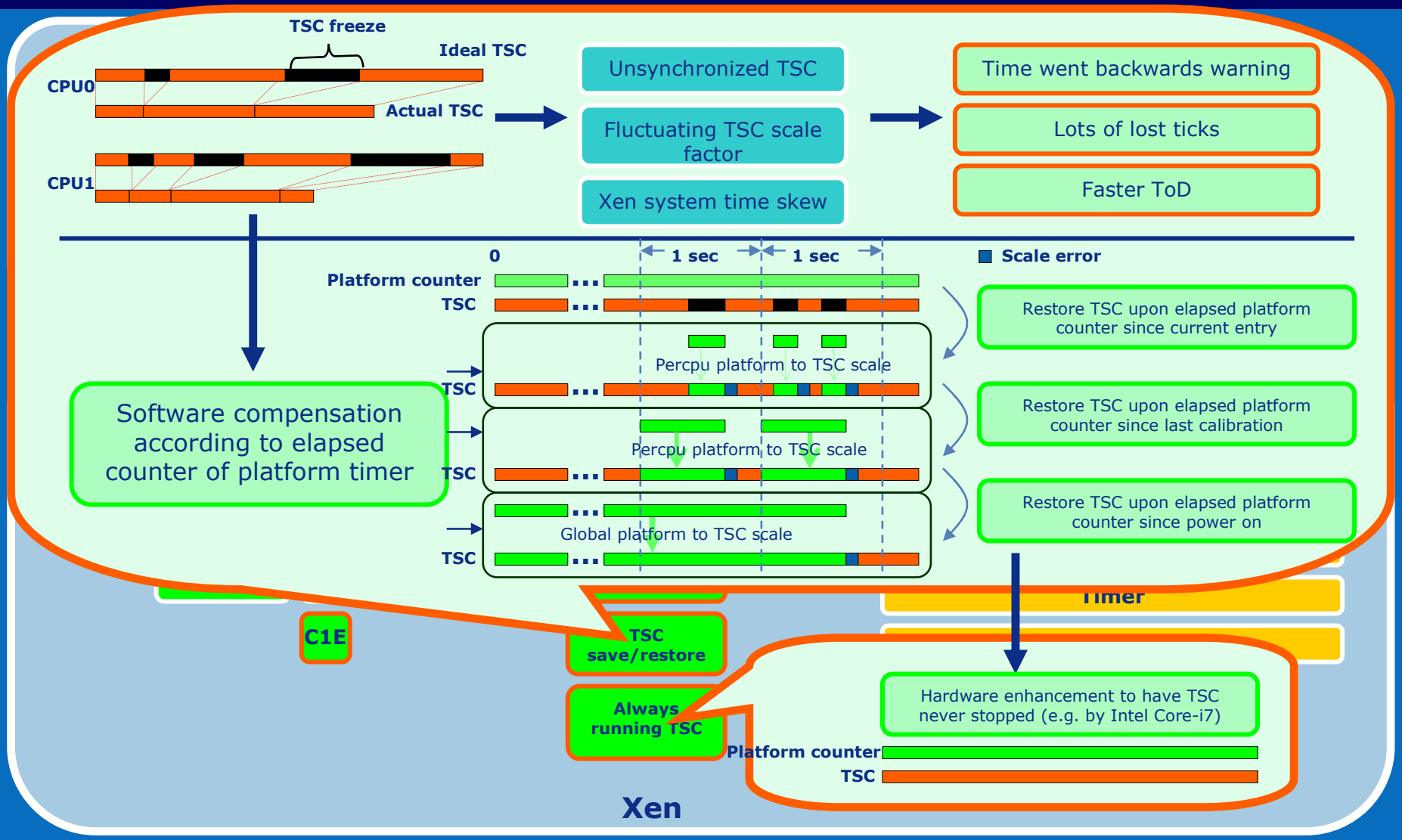
Enhanced C-states support



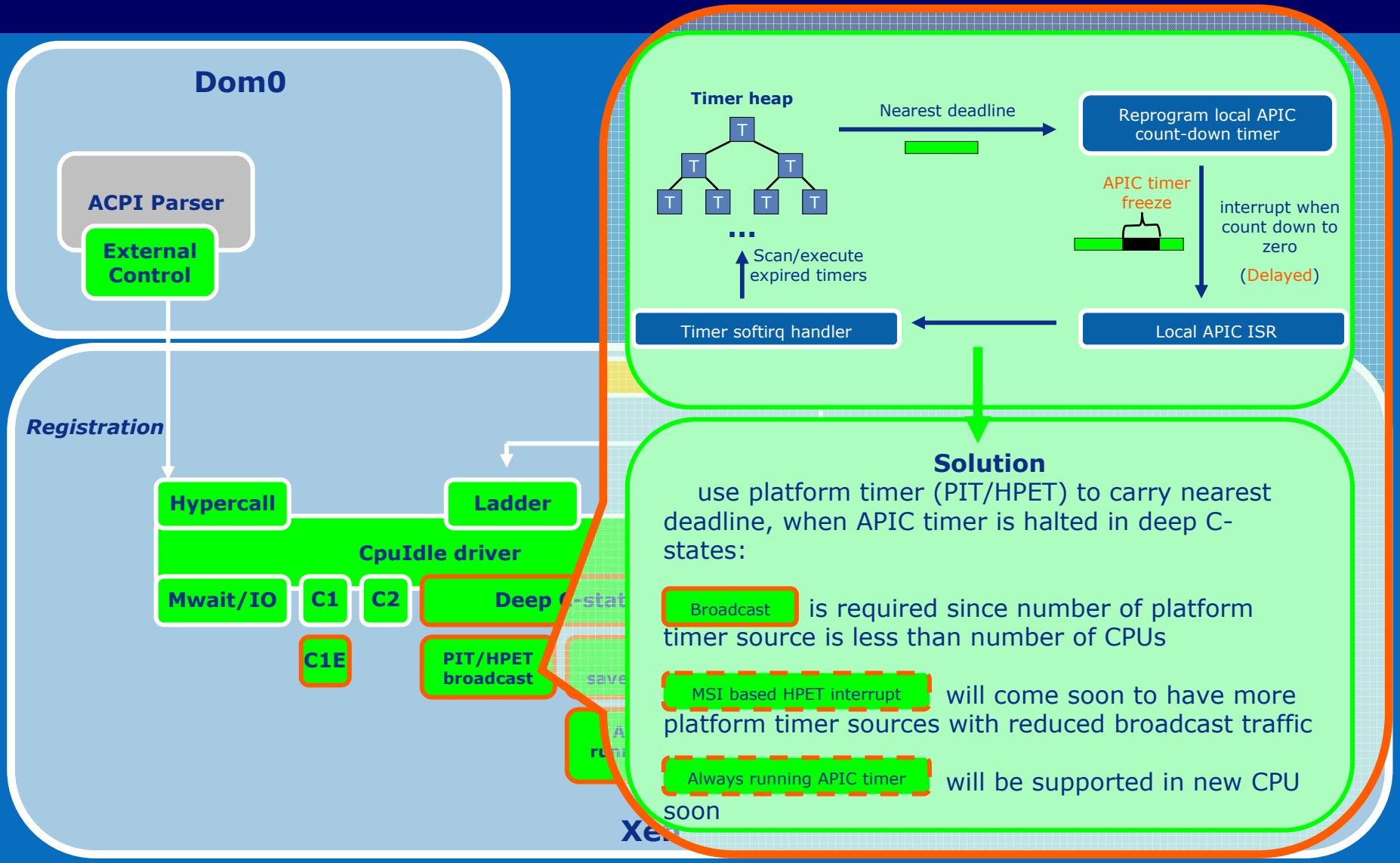
'noPM' has both cpuidle and cpufreq disabled, and vice versa for other two cases. Compared to 'C3', noC3 has maximum C-state limited to C2

For idle watt, lower value means greener. For SPECpower score, higher value indicates more power efficient

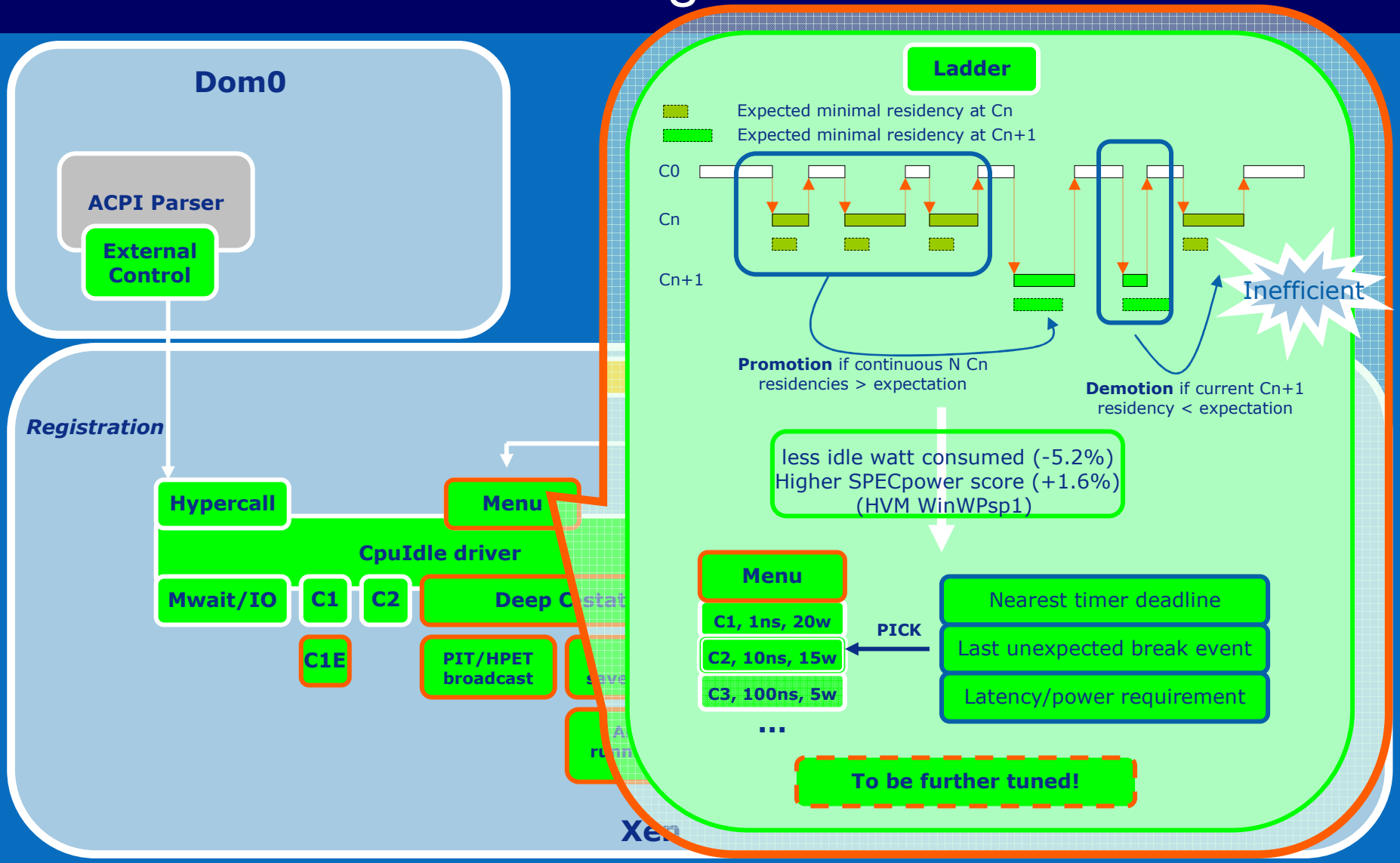
TSC freeze


TSC freeze
Ideal TSC
CPU0
Actual TSC
CPU1
Unsynchronized TSC
Fluctuating TSC scale factor
Xen system time skew
Time went backwards warning
Lots of lost ticks
Faster ToD
0
1 sec
1 sec
Scale error
Platform counter
TSC
Percpu platform to TSC scale
Percpu platform to TSC scale
Global platform to TSC scale
Restore TSC upon elapsed platform counter since current entry
Restore TSC upon elapsed platform counter since last calibration
Restore TSC upon elapsed platform counter since power on
Software compensation according to elapsed counter of platform timer
C1E
TSC save/restore
Always running TSC
Hardware enhancement to have TSC never stopped (e.g. by Intel Core-i7)
Platform counter
TSC
Xen

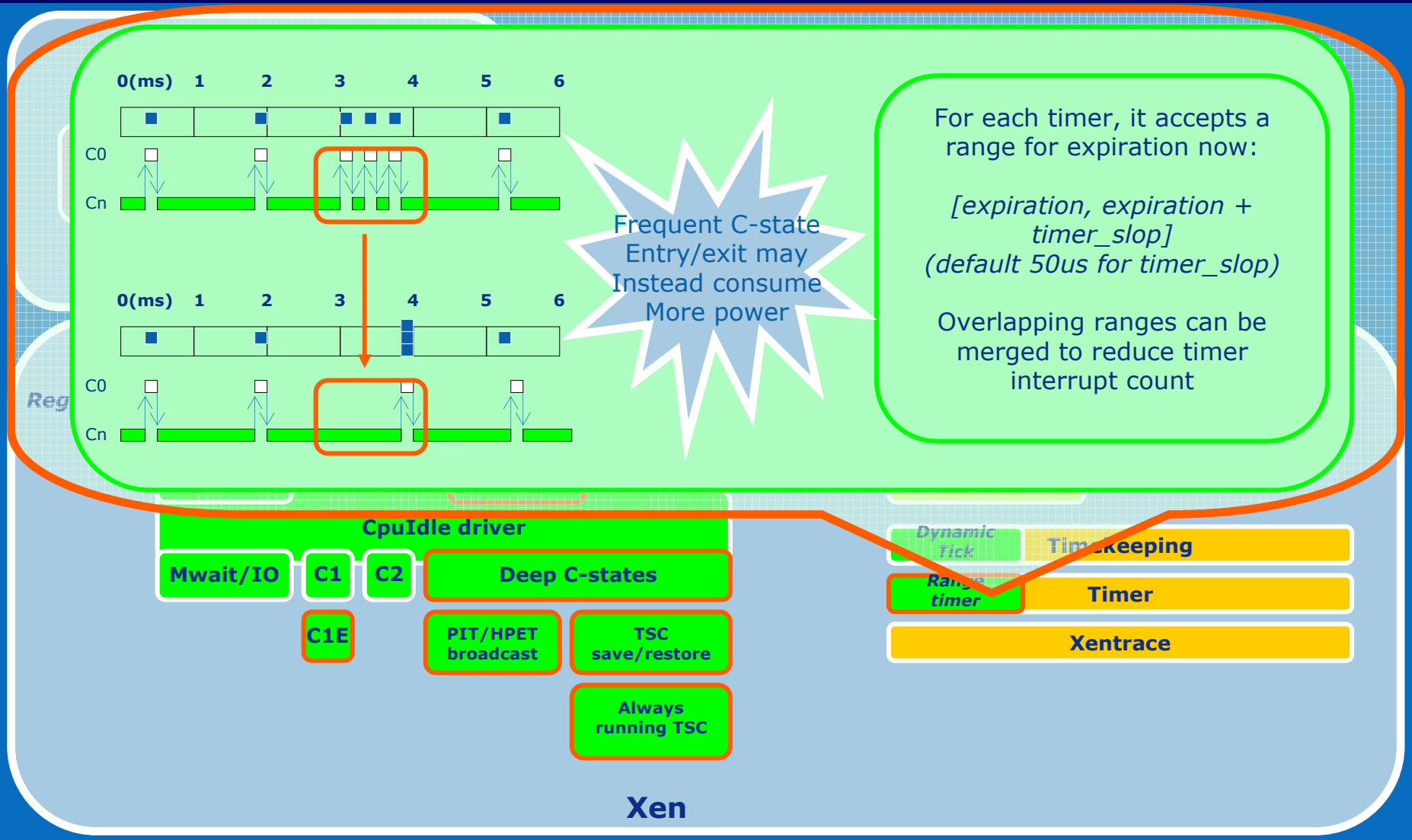
APIC timer freeze



Menu governor

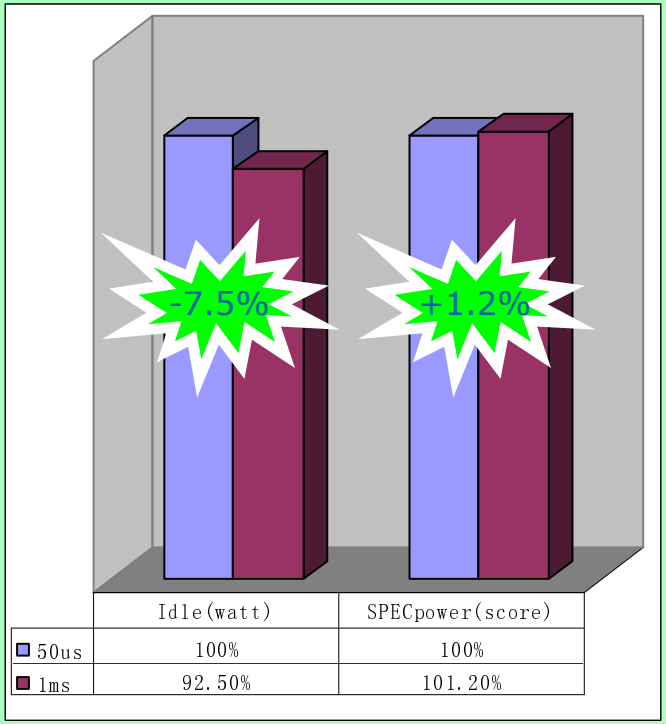


Range timer


0(ms) 1 2 3 4 5 6
C0
Cn
0(ms) 1 2 3 4 5 6
C0
Cn
Frequent C-state Entry/exit may Instead consume More power
For each timer, it accepts a range for expiration now:
[expiration, expiration + timer_slop]
(default 50us for timer_slop)
Overlapping ranges can be merged to reduce timer interrupt count
CpuIdle driver
Mwait/IO
C1
C2
Deep C-states
C1E
PIT/HPET broadcast
TSC save/restore
Always running TSC
Dynamic Tick
Timekeeping
Timer
Xentrace
Xen

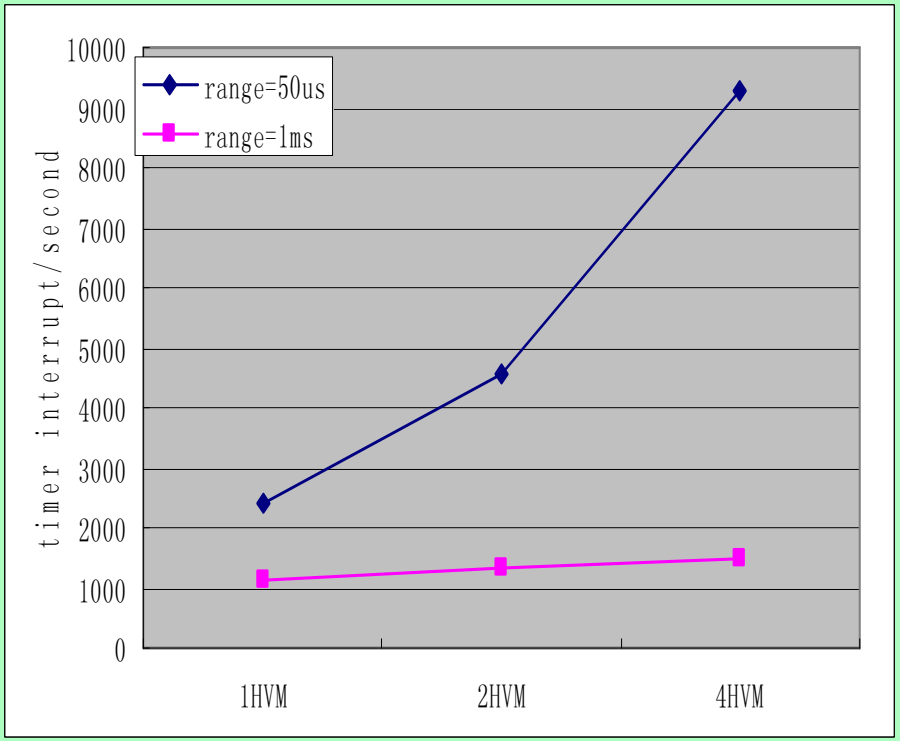
Range timer effect

One UP HVM RHEL5u1



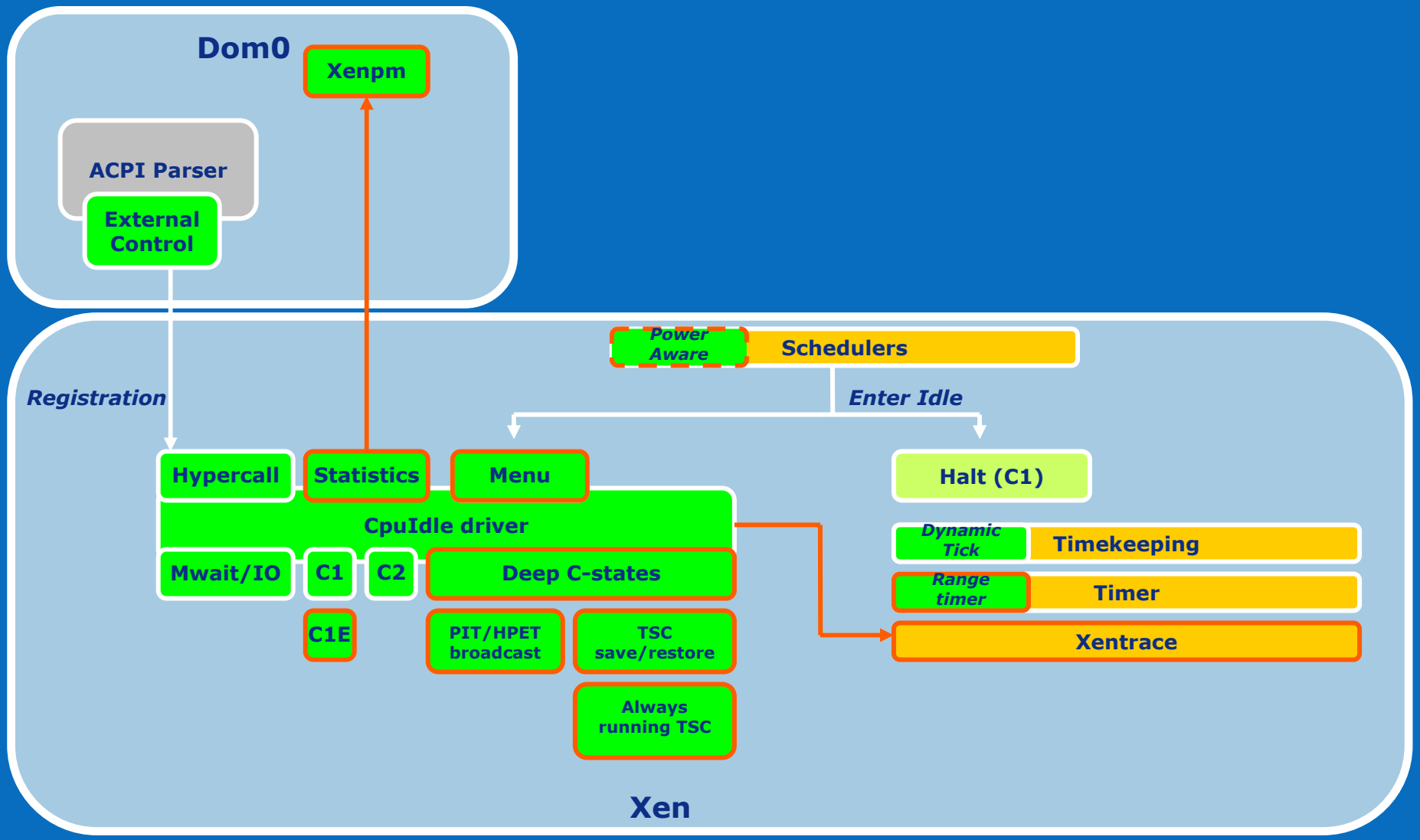
	Idle(watt)	SPECpower(score)
50us	100%	100%
1ms	92.50%	101.20%

Multiple idle UP HVM RHEL5u1



Collected on a two-cores mobile platform

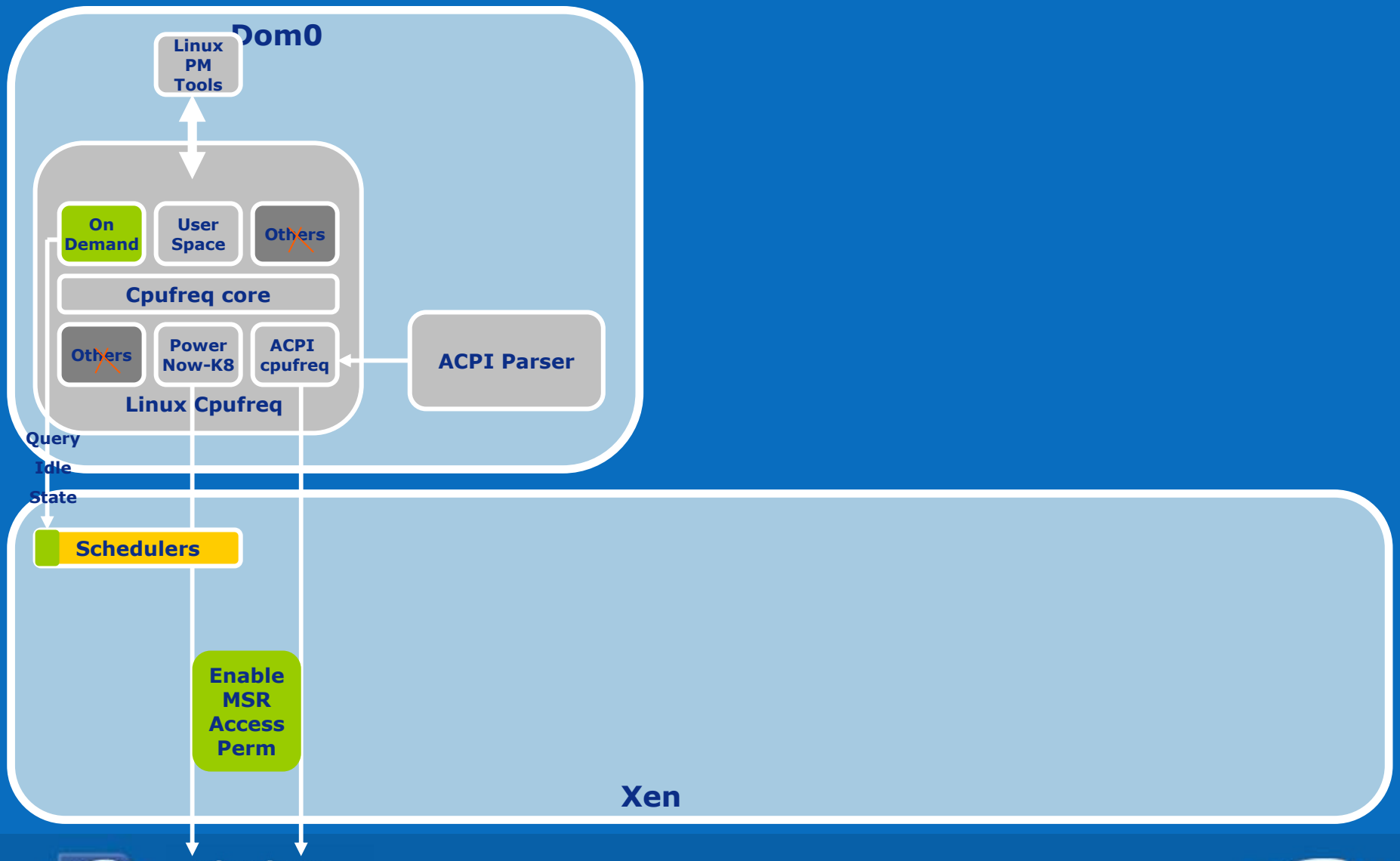
Current picture



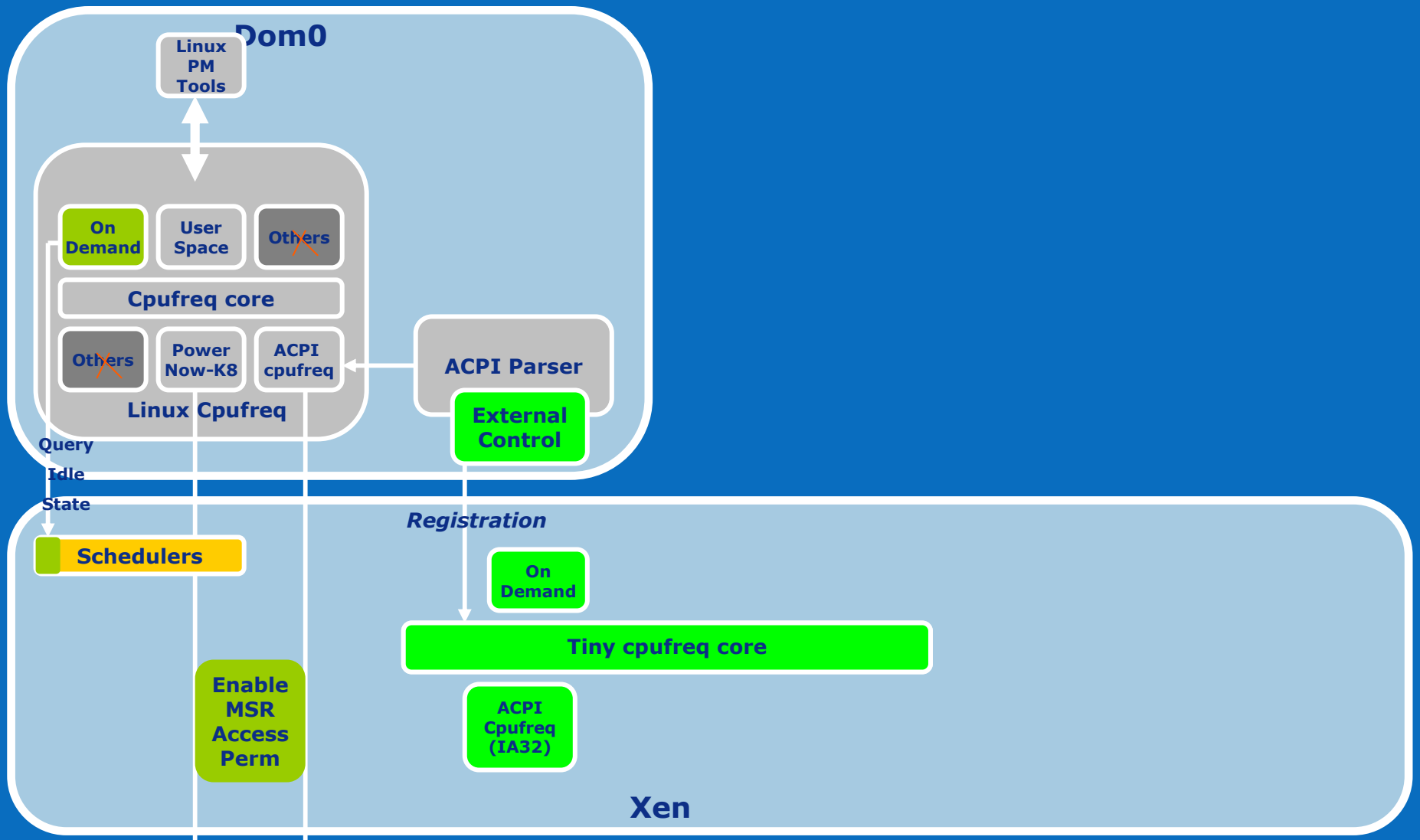
Evolutions of run-time power management

(Xen cpufreq)

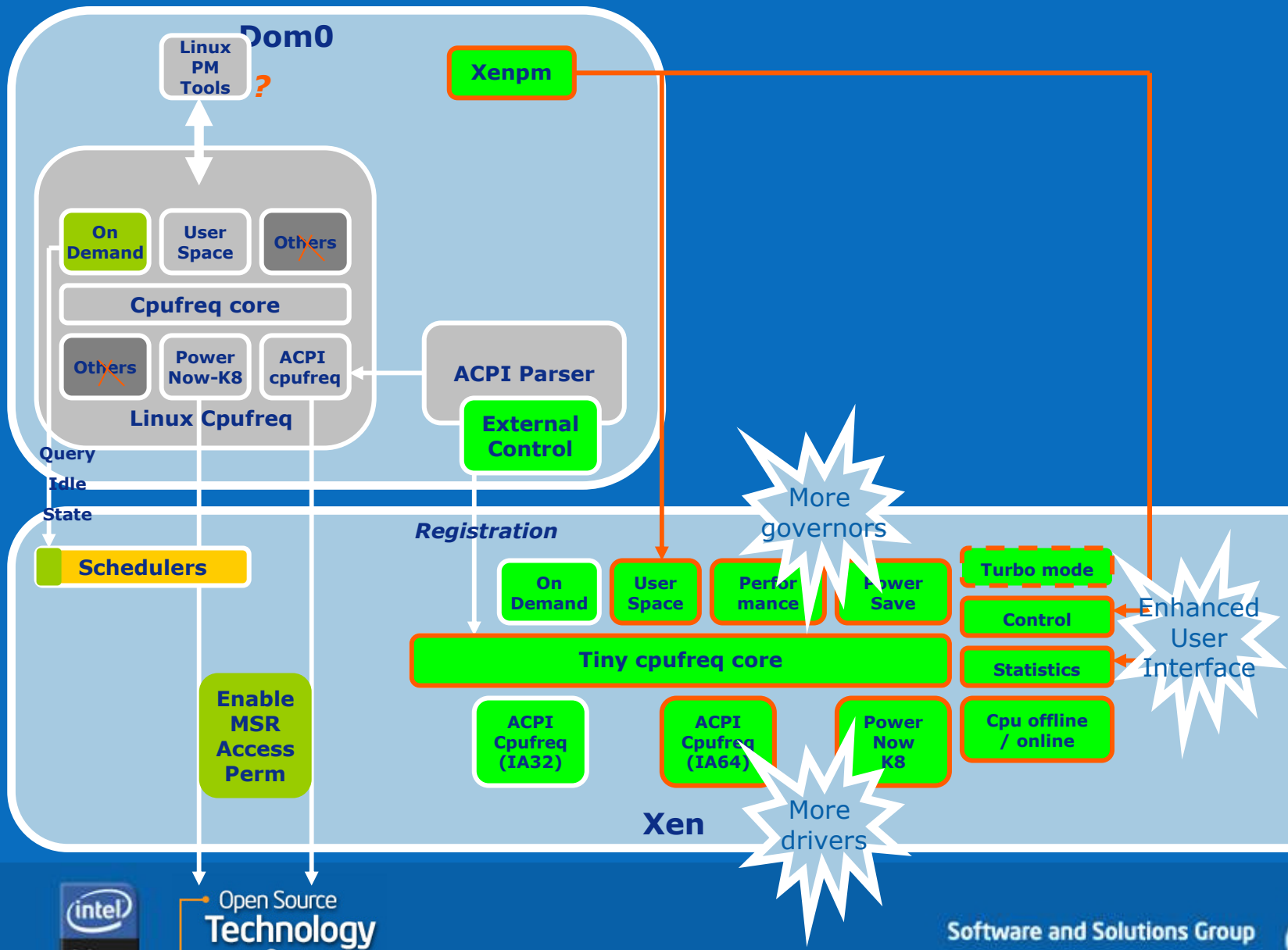
Xen 3.2


Dom0
Linux PM Tools
On Demand
User Space
Others
Cpufreq core
Others
Power Now-K8
ACPI cpufreq
ACPI Parser
Linux Cpufreq
Query Idle State
Schedulers
Enable MSR Access Perm
Xen

Xen Summit Boston 2008



Current picture



Tools

Dom0

Xenpm

Retrieve run-time statistics about Xen cpuidle and cpufreq

Apply user policy on exposed control knobs of Xen cpufreq (governor, set freq, etc)

More capabilities to be added later, e.g. profile…

```
[root@ktian1-dev ~]# xenpm start
Elapsed time (ms): 2883
CPU0:
        residency         percentage
  C0     1 ms     0.06%
  C1     0 ms     0.00%
  C2     2882 ms  99.94%
  P0     0 ms     0.00%
  P1     0 ms     0.00%
  P2     0 ms     0.00%
  P3     0 ms     100.00%

CPU1:
        residency         percentage
  C0     4 ms     0.16%
  C1     0 ms     0.00%
  C2     2879 ms  99.84%
  P0     0 ms     0.00%
  P1     0 ms     0.00%
  P2     0 ms     0.00%
  P3     2 ms     100.00%
```

Log every state change for Xen cpuidle and cpufreq:

```
CPU0  391365842416 (+   21204)  cpu_idle_entry  [ idle to state 2 ]
CPU0  391375951050 (+10108634)  cpu_idle_exit   [ return from state 2 ]
```

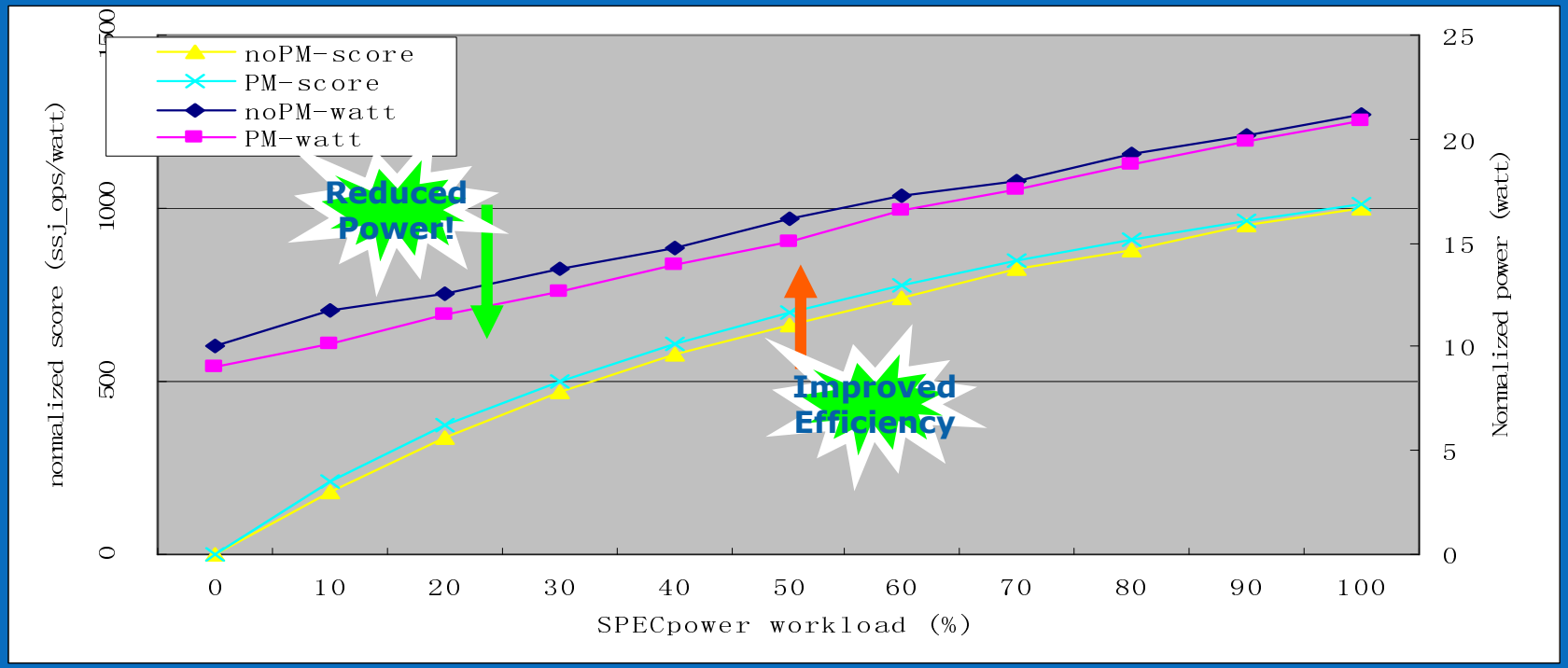
Raw data could be further processed by other scripts

Xentrace

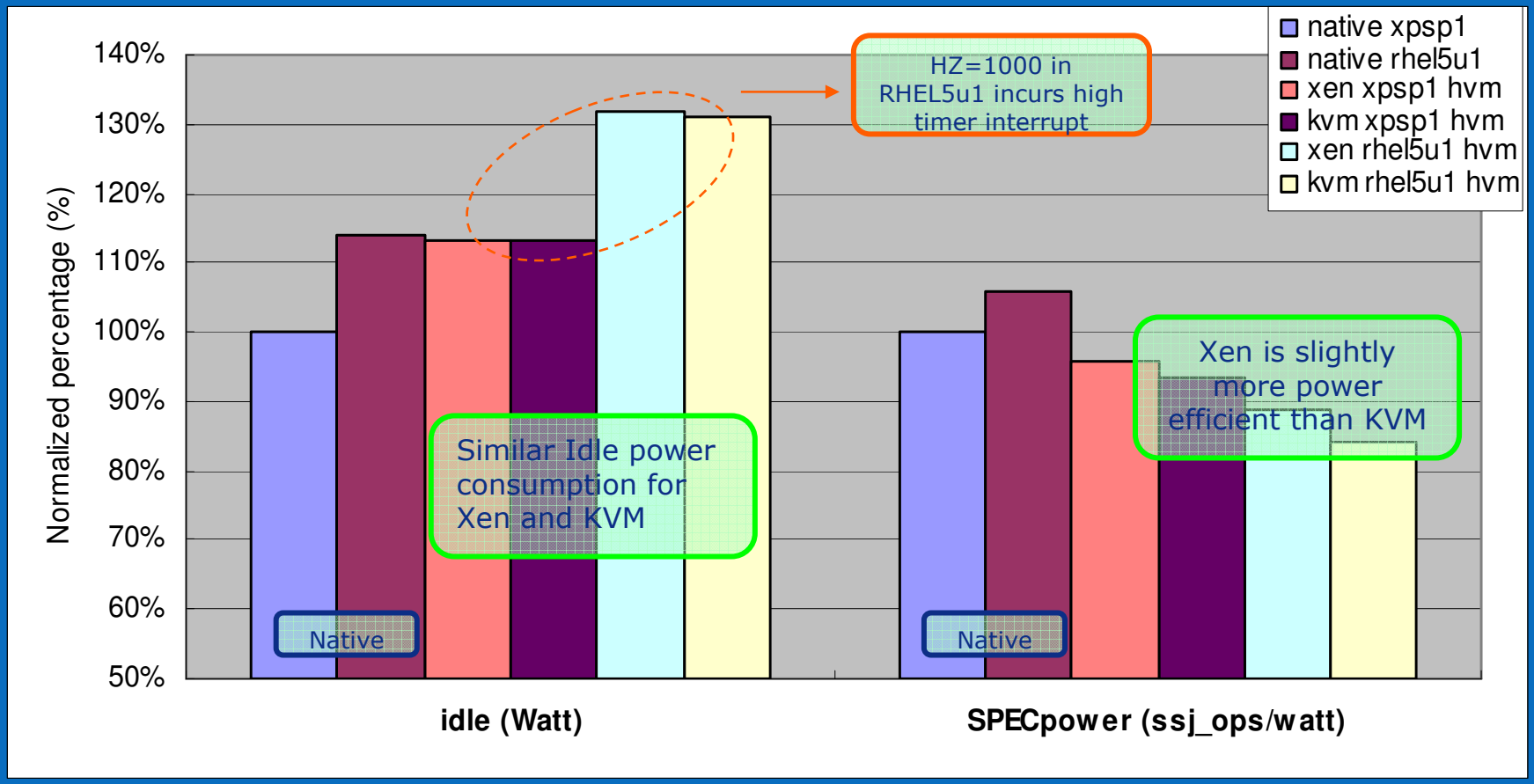
Xen

Experimental data about Xen power efficiency

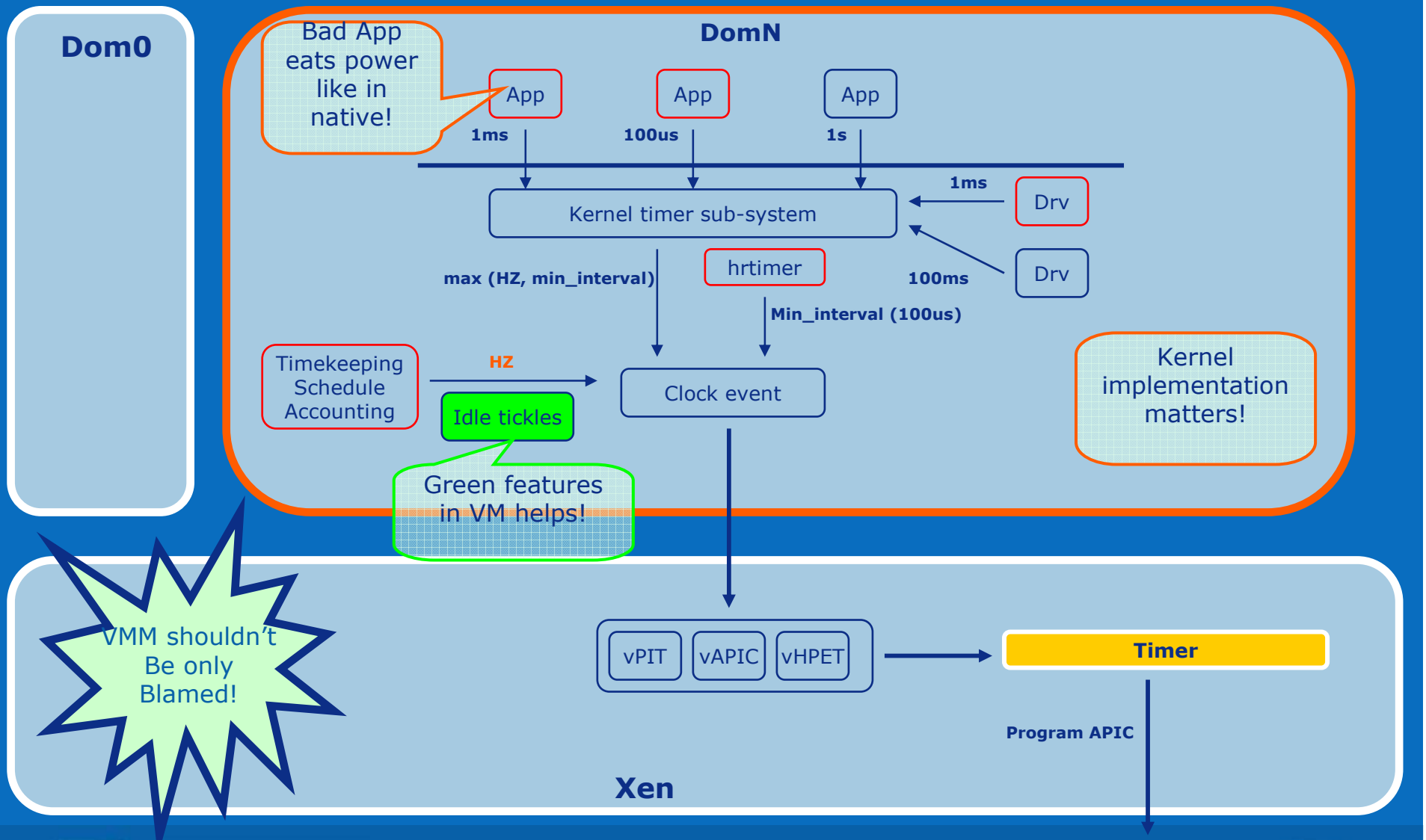
- All data shown in this section:
  - For reference only and not guaranteed
  - On a two-core mobile platform, with one HVM guest created
    - Server consolidation effect with multiple VMs/workloads are in progress
- Improvement when Xen cpuidle and cpufreq are enabled
  - SPECpower score is normalized (100% noPM score is 1000 ssj_ops / watt)
  - Similarly, consumed watt is also normalized (idle noPM watt is 10w)



- Below is a more attracting comparison
  - Native WinXPsp1's idle watt and SPECpower score are both normalized to 100% as the base

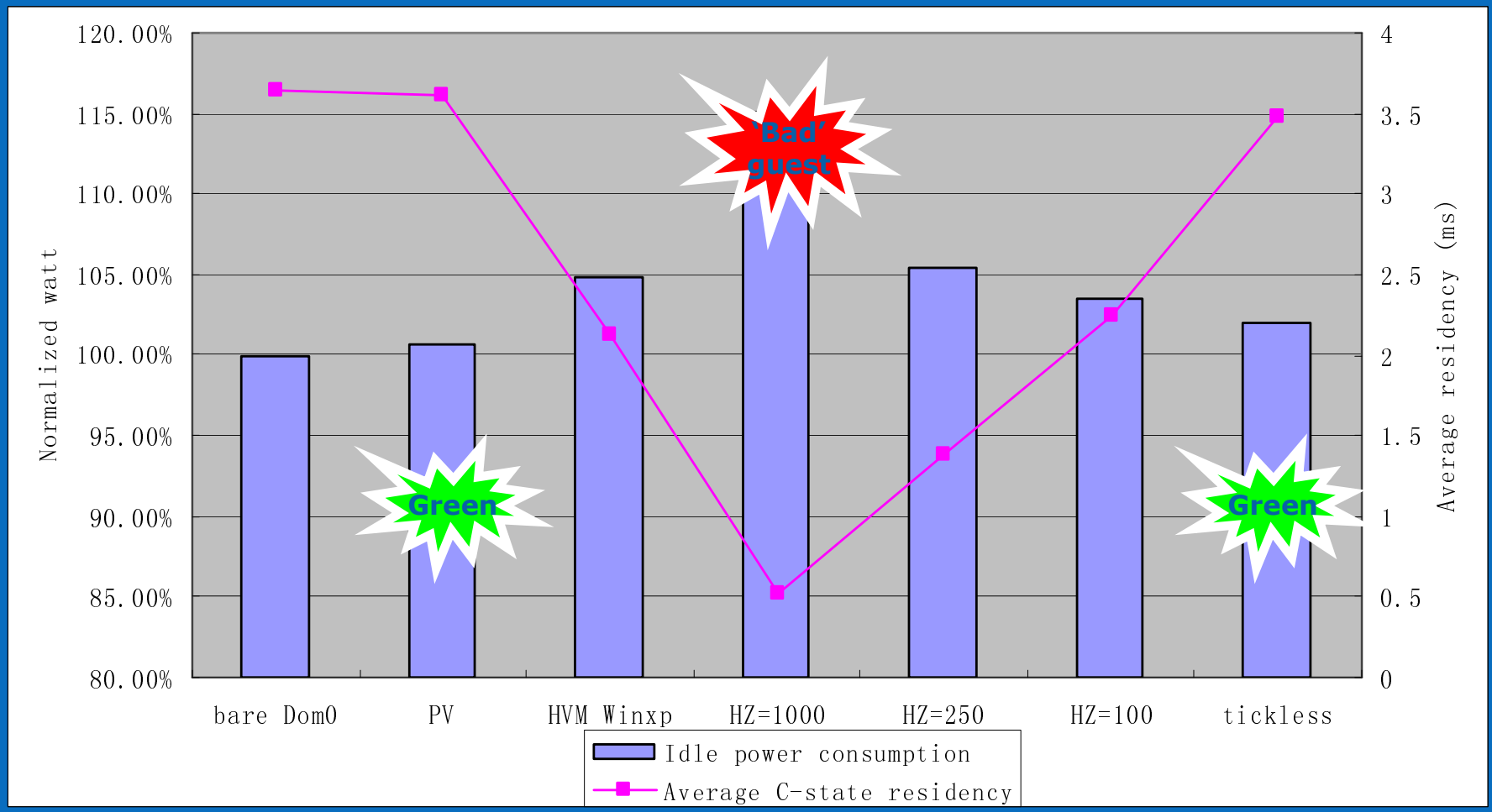


Power impact from VM

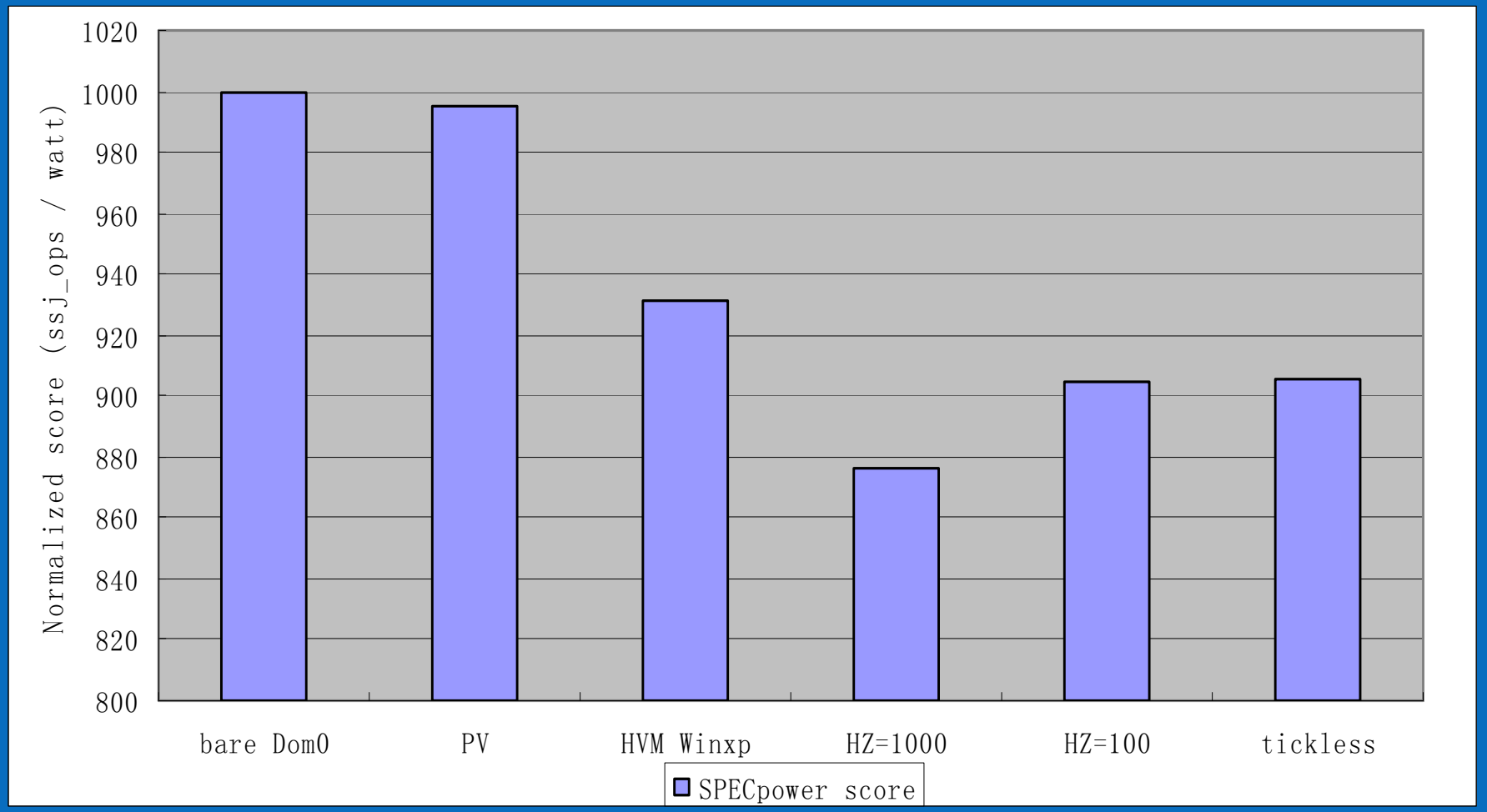

Dom0
DomN
Bad App eats power like in native!
App
App
App
1ms
100us
1s
1ms
Drv
Kernel timer sub-system
Drv
100ms
hrtimer
max (HZ, min_interval)
Min_interval (100us)
Timekeeping
Schedule
Accounting
HZ
Idle tickles
Clock event
Kernel implementation matters!
Green features in VM helps!
VMM shouldn't Be only Blamed!
vPIT
vAPIC
vHPET
Timer
Program APIC
Xen

- VMM shouldn't be blamed as only reason for high power consumption!
- A 'bad' VM could eat power
  - Just like what 'bad' application could do in a native OS
  - Cause high break events (e.g. timer interrupts) with short C-state residency
- Which parts in VM could draw high power?
  - 'Bad' applications hurt just as what they could on native
  - How guest OS is implemented also matters
    - Periodic tick frequency – HZ
    - Timer usage in drivers
    - Time sub-system implementation
    - …
- Green guest OS wins!
  - Smaller HZ, idle tickles or fully dynamic tick, range timer, etc.

Idle power consumption



Power efficiency



Legal Information

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. 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 may make changes to specifications, product descriptions, and plans at any time, without notice.
- All dates provided are subject to change without notice.
- Intel is a trademark of Intel Corporation in the U.S. and other countries.
- *Other names and brands may be claimed as the property of others.
-


intel
Leap ahead™