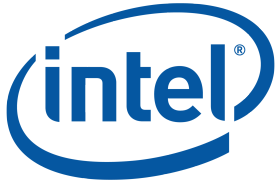


Linux* Storage Power consumption

Arjan van de Ven
Kristen Carlson Accardi

Open Source Technology Center

Legal Information

Summary of work in 2007

ALPM – Aggressive Link Power Management
- Saves average of .6 Watts per disk when in min_power mode
- In 2.6.24

AN – Asynchronous Notification
- Allows media change notifications to user space
- Prevents the need for polling

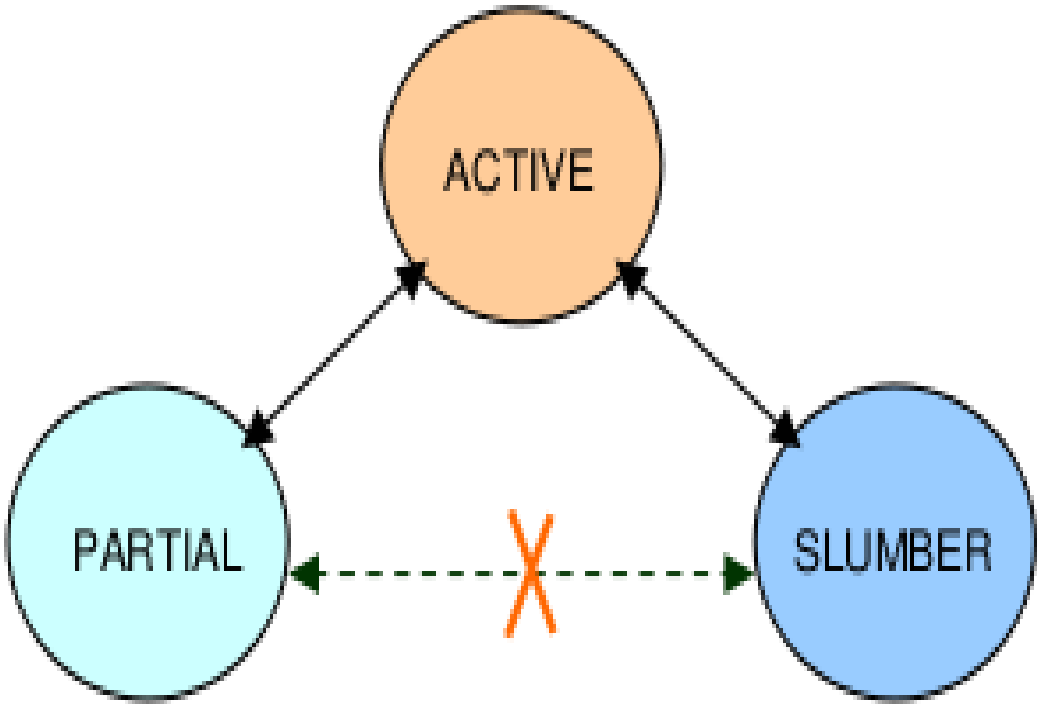
Application improvements through PowerTOP visibility
- Draws attention to behavior which keeps system busy unnecessarily
  - It's hard to pinpoint disk activity because all we get is kjournald and pdflush...

What is ALPM?

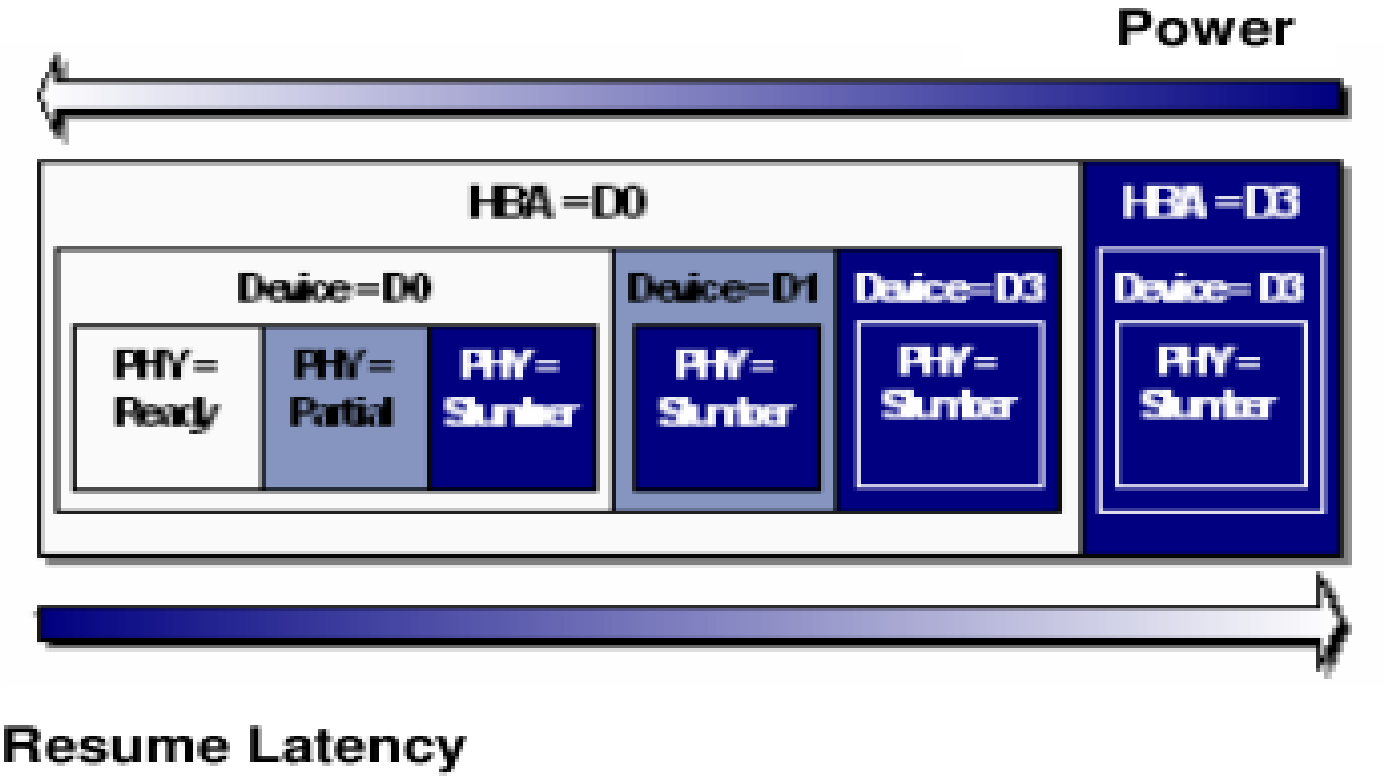
Aggressive Link Power Management (ALPM)

This feature is described in detail in the AHCI 1.x spec. ALPM is a power-saving technique that focuses on the SATA link. When enabled, it allows the host controller and the disk to negotiate when to lower the power of the SATA link. When it is enabled, it can provide power savings of anywhere from .5-1.5 Watts per disk, depending on the system.

ALPM State Transitions



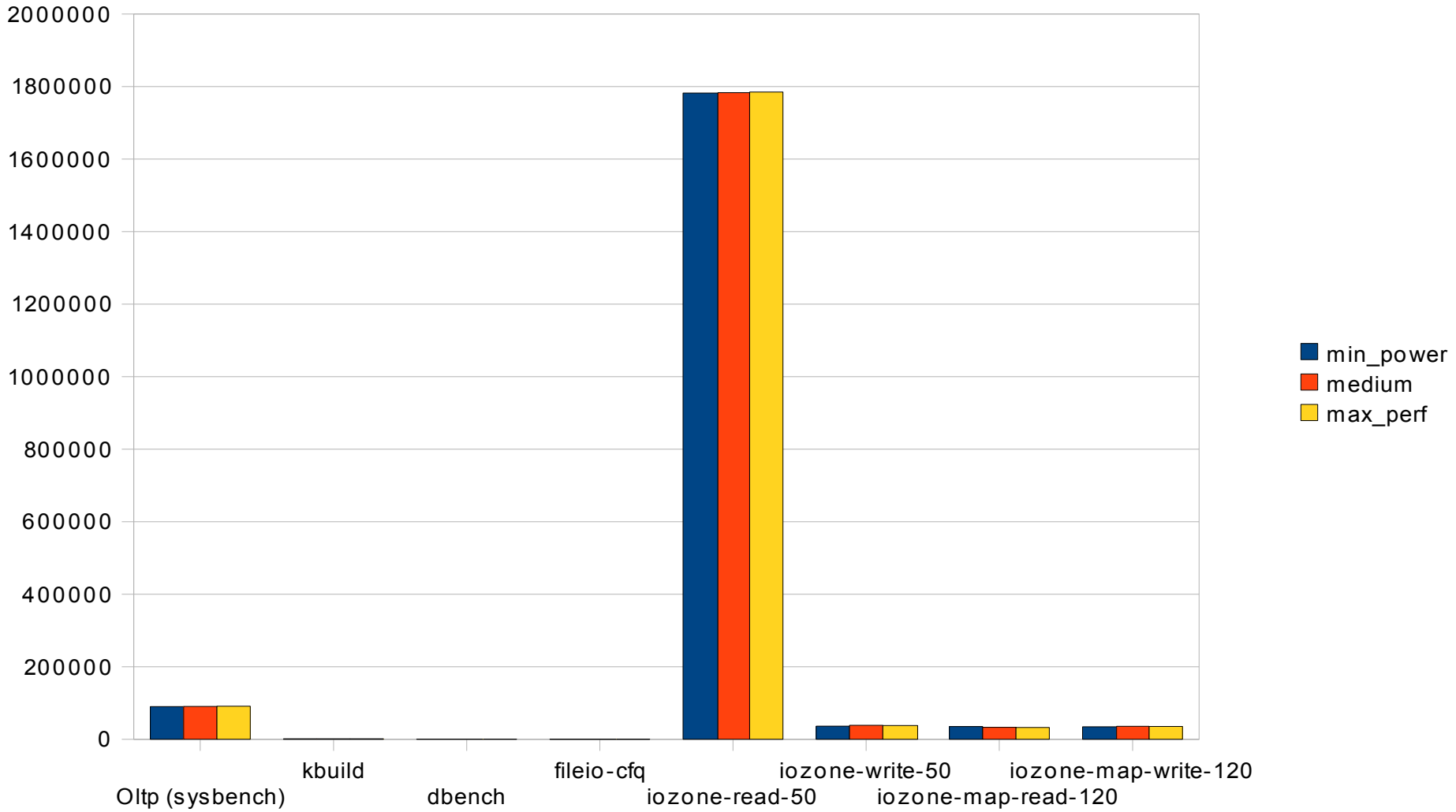
SATA Link and Device Power Management



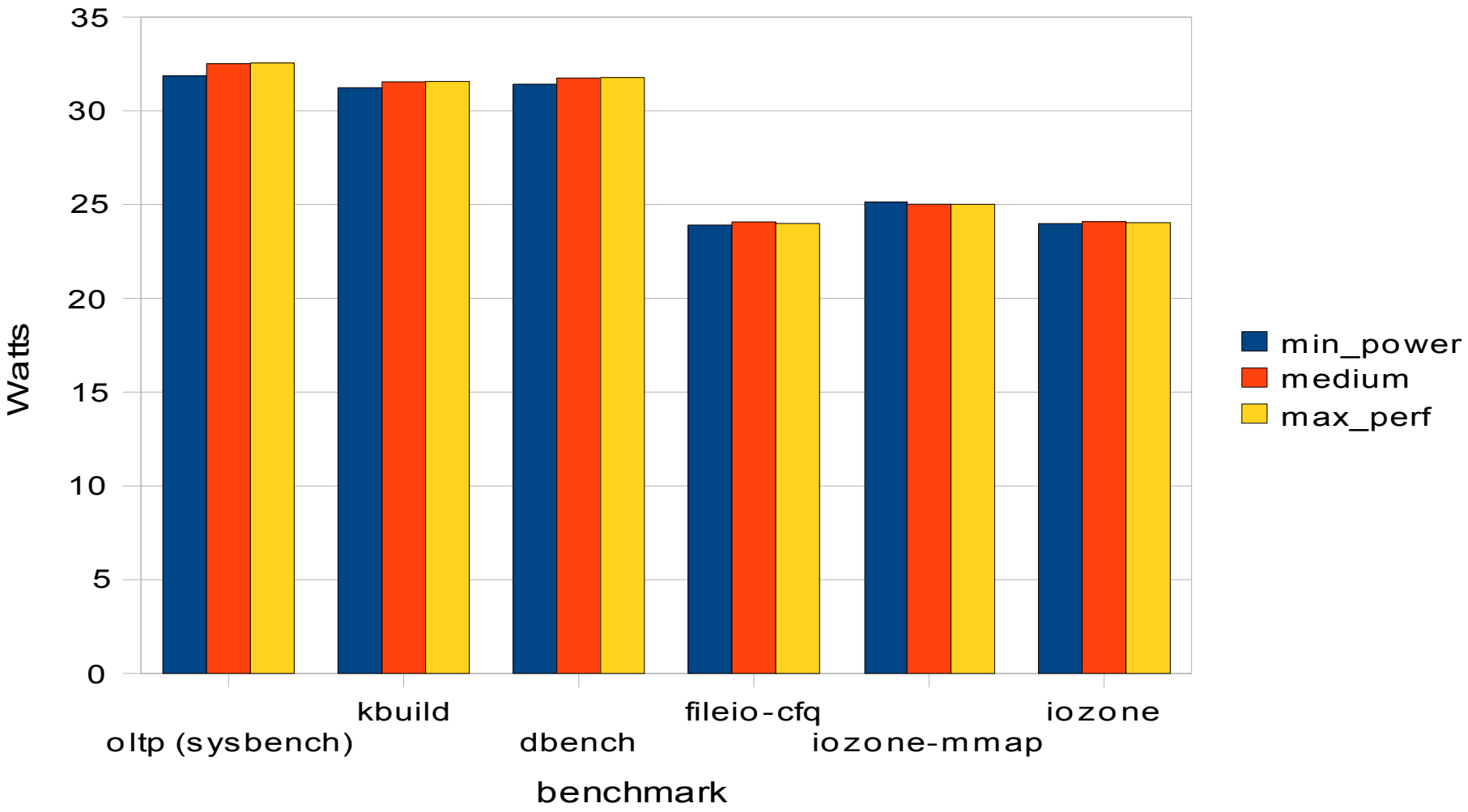
ALPM vs. performance

- Standard benchmarks show no performance degradation with ALPM enabled because they tend to keep the disk busy
- Bursty workloads may show a performance hit due to latency coming out of SLUMBER.
  - Latency to go from SLUMBER -> ACTIVE is 10 milliseconds
- PARTIAL mode is recommended, because it is the best power/performance trade off.
  - Latency to go from PARTIAL->ACTIVE is 10 microseconds
  - Power savings reduced by about 50% over SLUMBER

Throughput/Runtime



Average Power Consumption



Future work

Suggestion #1: Batch and Group I/O

Problem:

Delays and ripples wake CPU out of lower sleep state during DMA. Memory will also go out of lower sleep state, as well as ALPM enabled disks.

Suggestion:

- Batch I/O per disk instead of per process.
- Have laptop mode on by default
  - There is no known performance penalty for this, so why is it not the default?

Suggestion #2: Driver Hints

Problem:

Performance or latency issues make lower power modes less attractive for some workloads.

Drivers may not know when we are going to be active due to plugging, but block driver could provide advance notice.

Suggestion:

Have block layer provide callbacks to driver layer for 'active' and for 'idle'. This would allow subsystems other than SATA to provide their own implementations for low power mode. For example, USB can suspend.

Suggestion #3: Smarter timers

Problem:

Every time a timer fires, it wakes up the CPU

Suggestion:

Use a timer struct that accepts a range of values so that the scheduler could batch timer interrupts.

- What kind of ranges work?

Grouping timers