

# Scheduling on an Asymmetric Multicore System

---

Ren Wall, Nic Rust, Gavin Austin

12 Dec 2019

# Asymmetric Multicore systems.

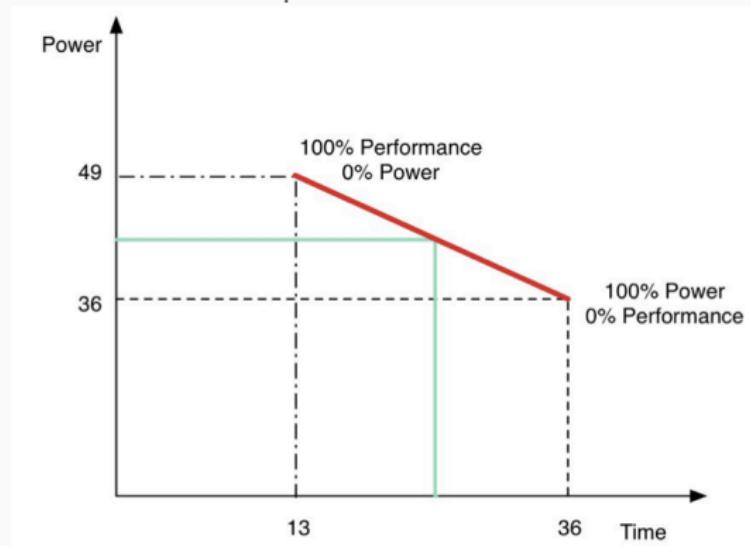
ARM big.LITTLE systems

Large X86-64 CPUs with different speed cores.



# Optimization Goals [Ant13]

Computation time  
Power Consumption



## Energy Aware Scheduler (EAS)[Kuc15]

EAS is the first Linux implementation of the CFS with built in idling and frequency tuning.

Implemented in 2015.

Greatly improves upon previous iteration.

[Lin15]

# Improvements Over Previous Scheduler[Lin15]

cpuidle vs sched-idle  
cpufreq vs sched-freq

## Drawbacks of EAS

If all cores are at full capacity its just a CFS.

# WASH Compared to EAS

Wash always schedules on highest power core.

Thread priorities arranged by most contended held locks.

Most waited on thread is placed on fastest core.

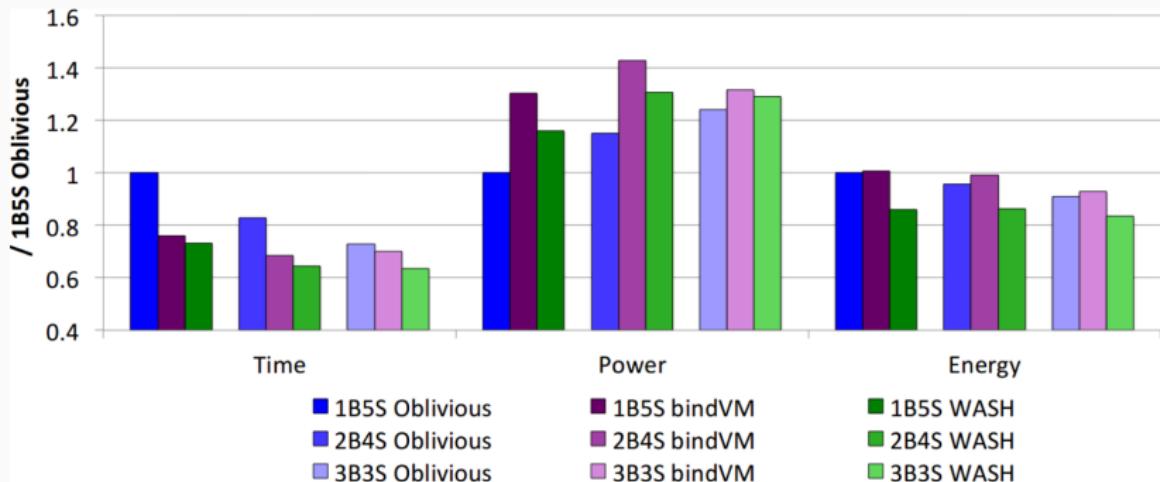


Figure 1: CFS (Oblivious), manual core prio (bindVM); 1B5S (1Big core and 5 Small cores), 2B4S and 3B3S [JCBM16].

# WASH Scheduler

---

## Algorithm 1 WASH

---

```
function WASH( $T_A, T_V, C_B, C_S, t$ )
     $T_A$ : Set of application threads
     $T_V$ : Set of VM service threads,  $T_A \cap T_V = \emptyset$ 
     $C_B$ : Set of big cores
     $C_S$ : Set of small cores,  $C_B \cap C_S = \emptyset$ 
     $t$ : Thread to schedule where  $t \in T_A \cup T_V$ 
    if  $|T_A| \leq |C_B|$  then
        if  $t \in T_A$  then Set Affinity of  $t$  to  $C_B$ 
        else Set Affinity of  $t$  to  $C_B \cup C_S$ 
        end if
    else if  $t \in T_A$  then
        if  $\forall \tau \in T_A (\text{Lock\%}(\tau) \leq \text{Lock}_{\text{Thresh}})$  then
            Set Affinity of  $t$  to  $C_B \cup C_S$ 
        end if
    end if
end function
```

-  Pantelis Antoniou, *Adventures in (simulated) asymmetric processing*, 2013.
-  Ivan Jibaja, Ting Cao, Stephen M. Blackburn, and Kathryn S. McKinley, *Portable performance on asymmetric multicore processors*, Proceedings of the 2016 International Symposium on Code Generation and Optimization (New York, NY, USA), CGO '16, ACM, 2016, pp. 24–35.
-  Amit Kucherla, *Energy aware scheduling (eas) project*, 2015.
-  Linaro, *Energy aware scheduling (eas) progress update*, 2015.