

Proposing a Replacement for FreeBSD's powerd

Or, how I tamed the fan of my notebook

Dominic Fandrey

EuroBSDcon 2016, September 2016

kamikaze a.k.a. Dominic Fandrey

- ▶ Dominic Fandrey <kami@freebsd.org>
- ▶ M.Sc. (Computer Science)
- ▶ Located in Europe/Karlsruhe



Contents

A long long time ago (February)

Where load stats come from

Sampling vs. Filtering

Load Aggregation

The Nail in the Coffin

Per Core Clock Frequencies

fin

attachments



A long long time ago (February)

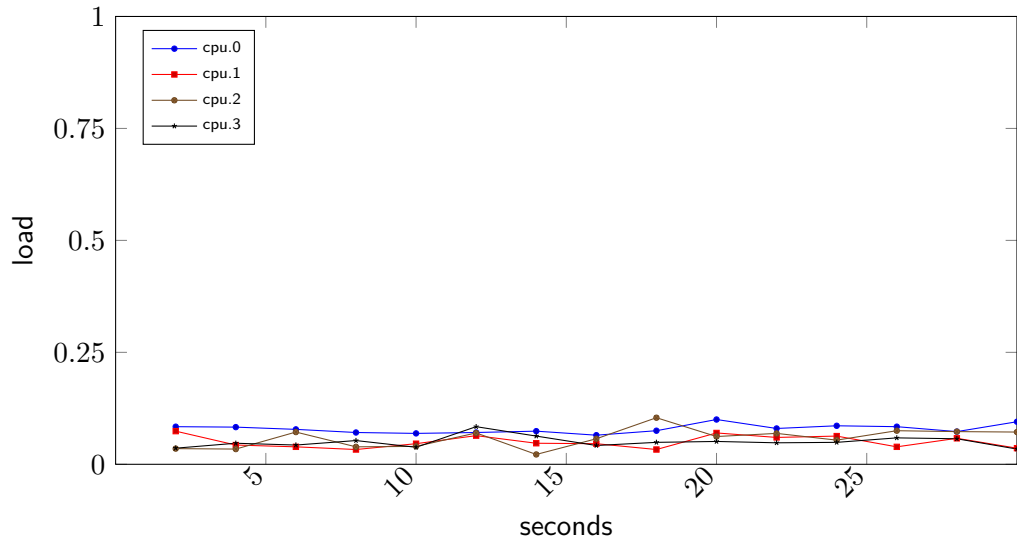


A long long time ago (February)

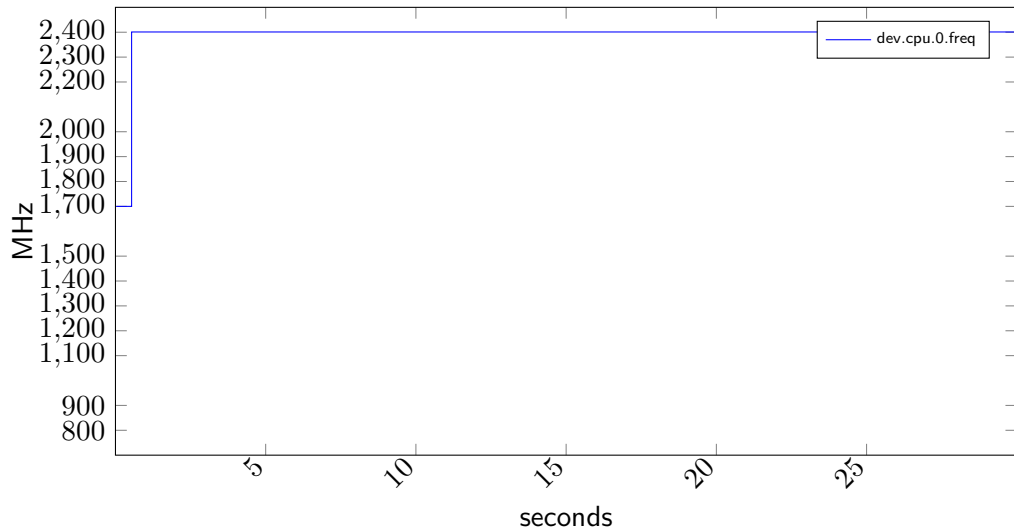
- ▶ github.com/lonkamikaze/powerdxx
- ▶ lonkamikaze.github.io/tags/#tag-power-management
- ▶ sysutils/powerdxx



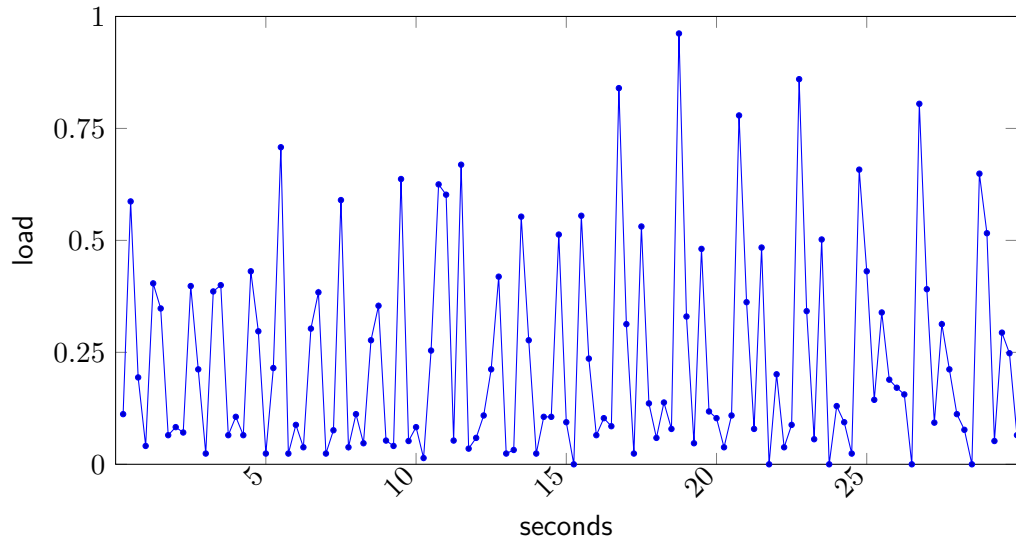
browser/email load according to *top*



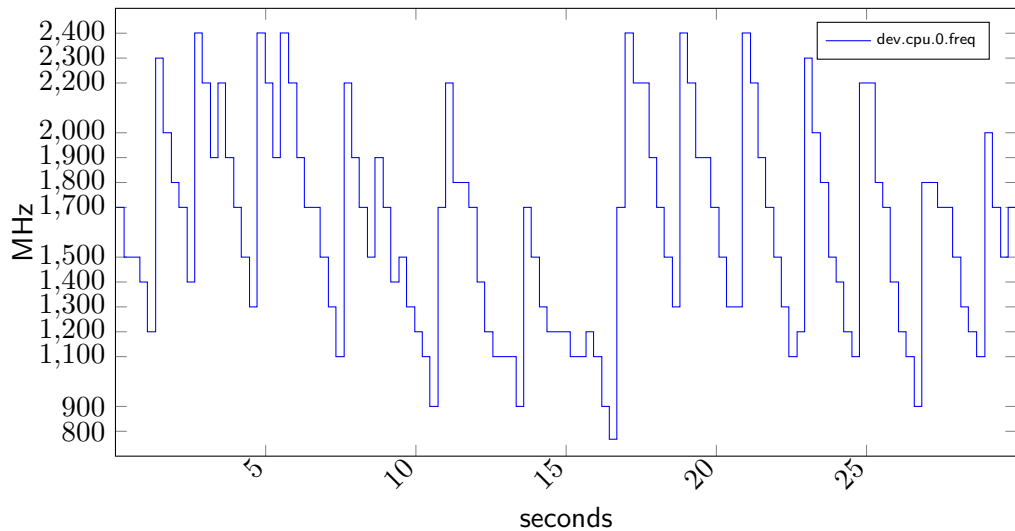
browser/email clock frequency running *powerd*



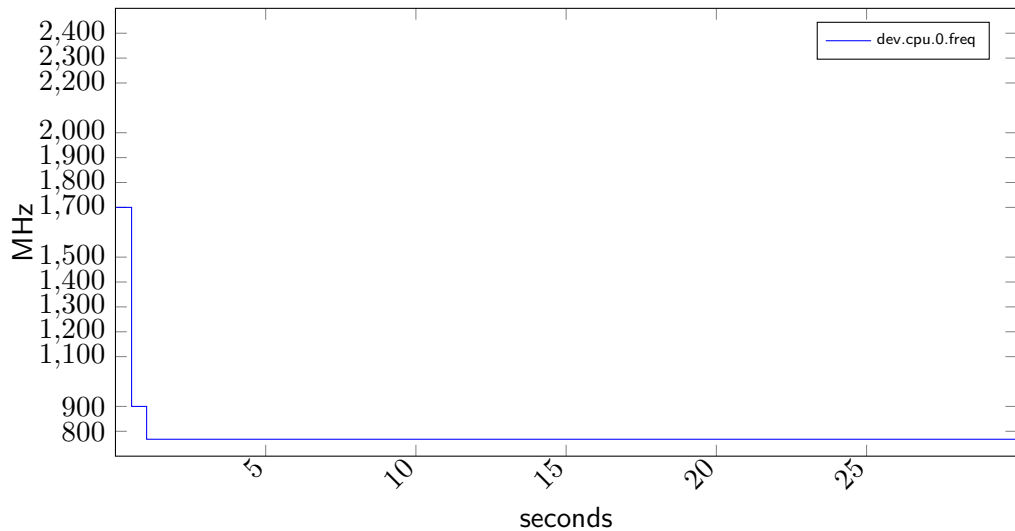
browser/email load according to *powerd*



browser/email clock frequency running *powerd adaptive*



browser/email clock frequency *expectation*



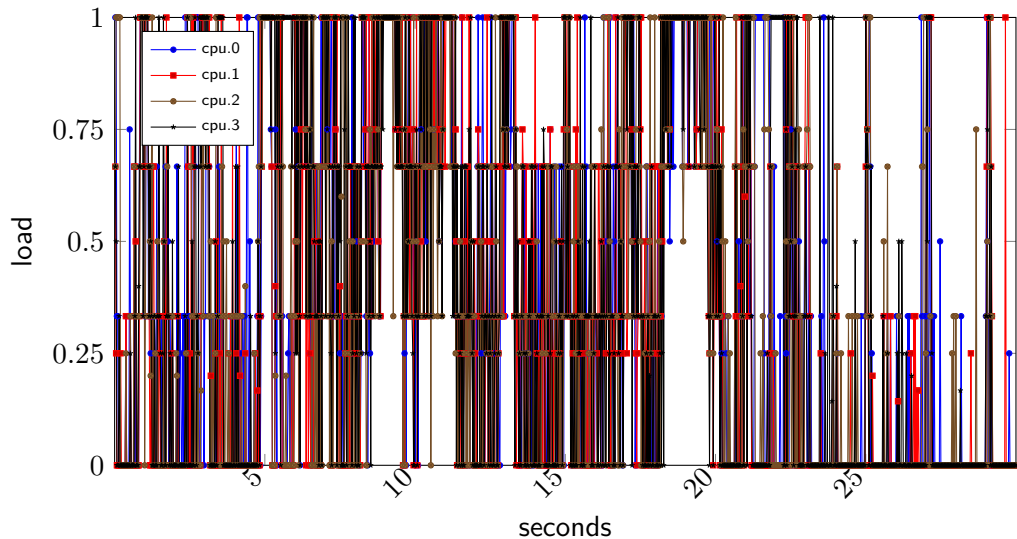
Where load stats come from



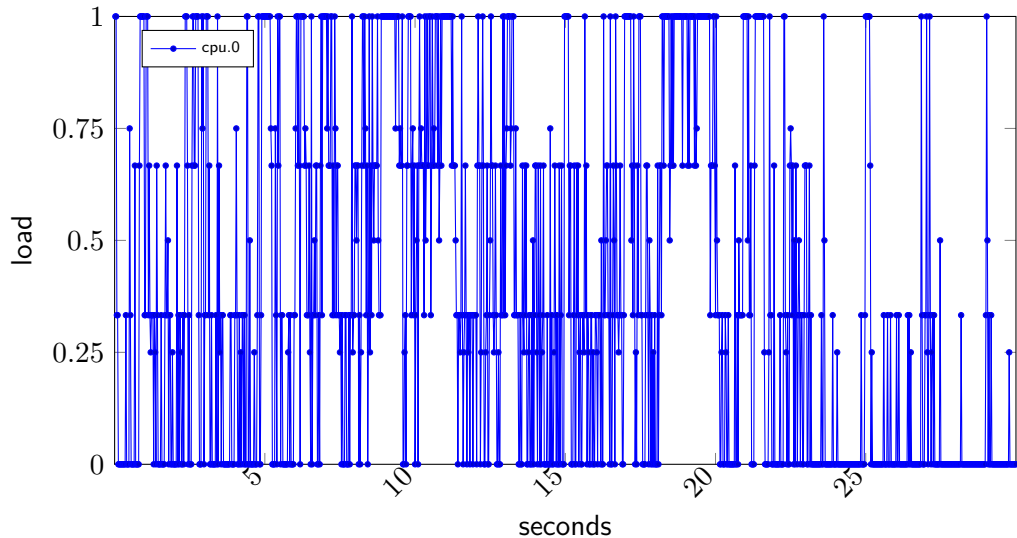
Sampling vs. Filtering



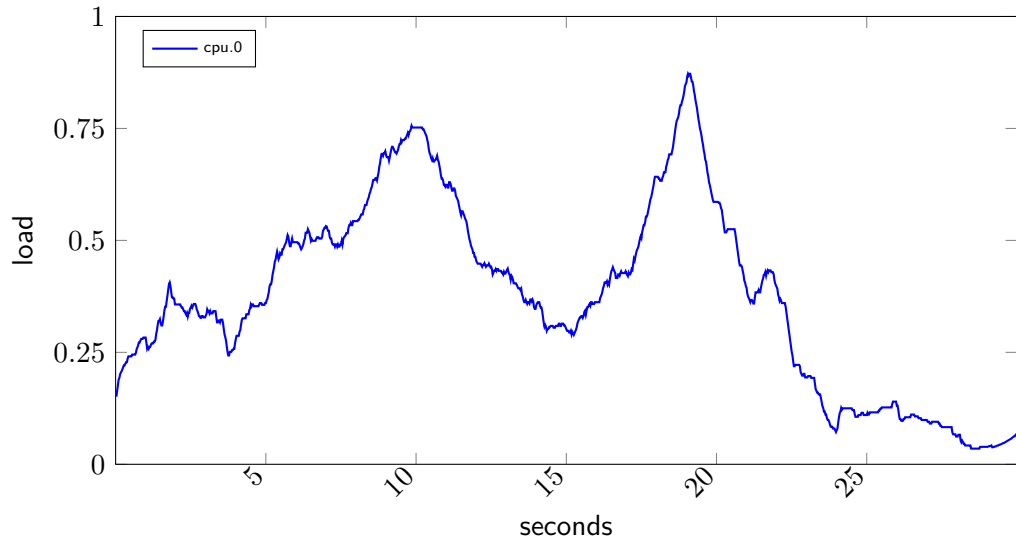
load 25 ms sampling



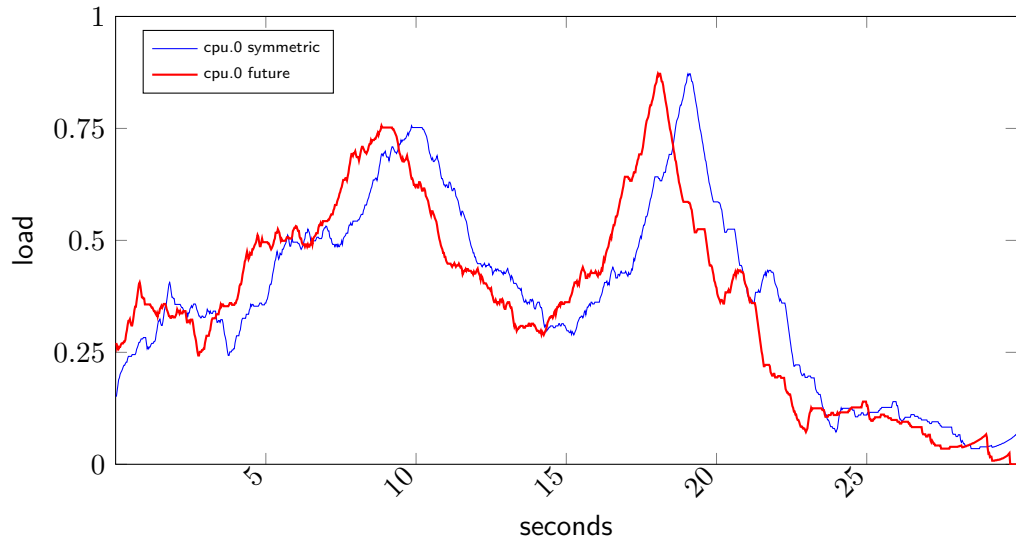
load 25 ms sampling



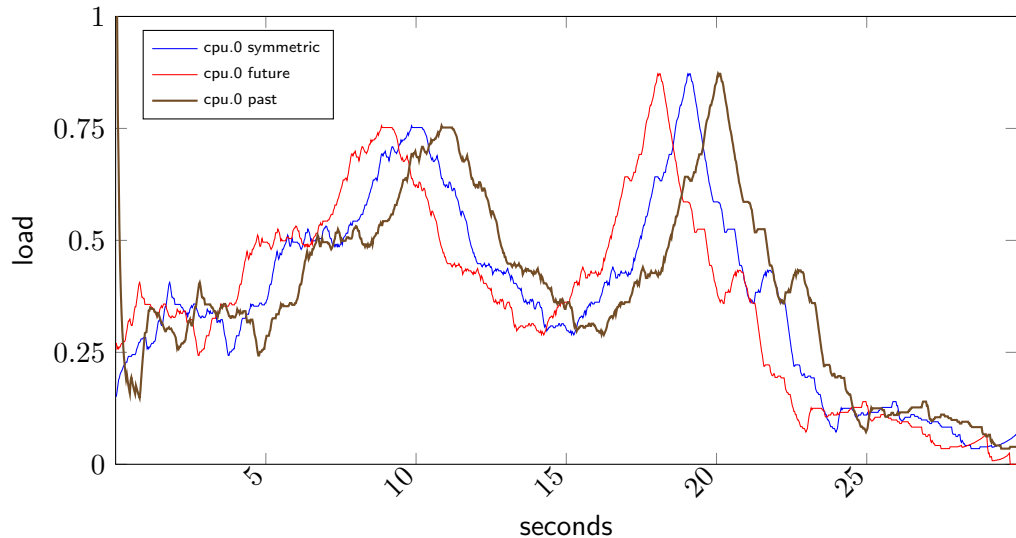
load 25 ms sampling, 2 s filter



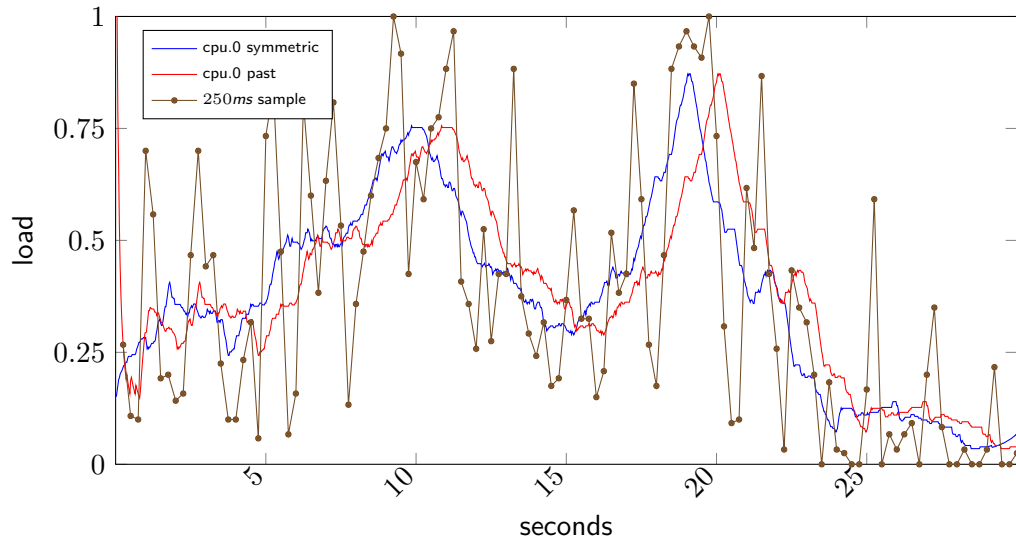
load 25 ms sampling, 2 s filter



load 25 ms sampling, 2 s filter



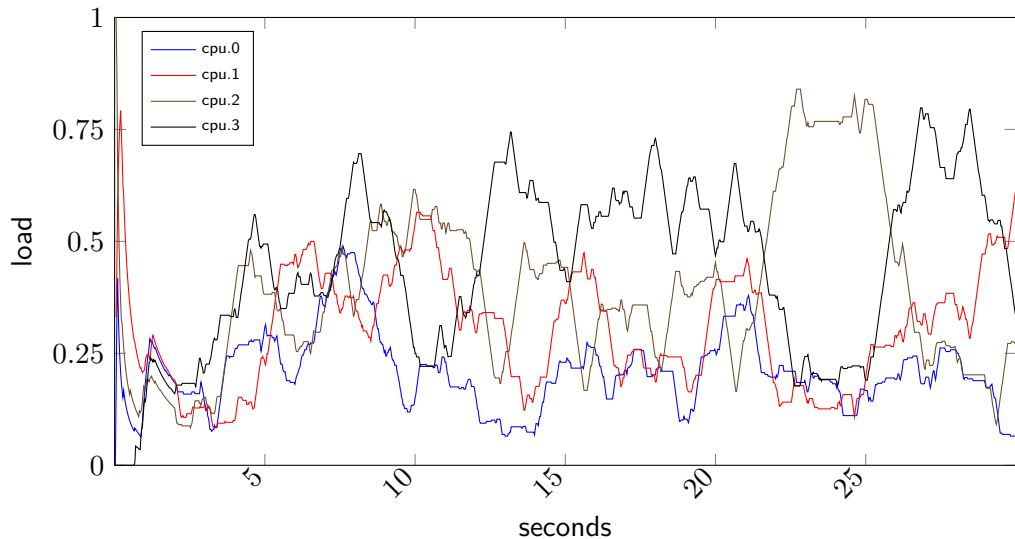
load sampling vs. filtering comparison



Load Aggregation



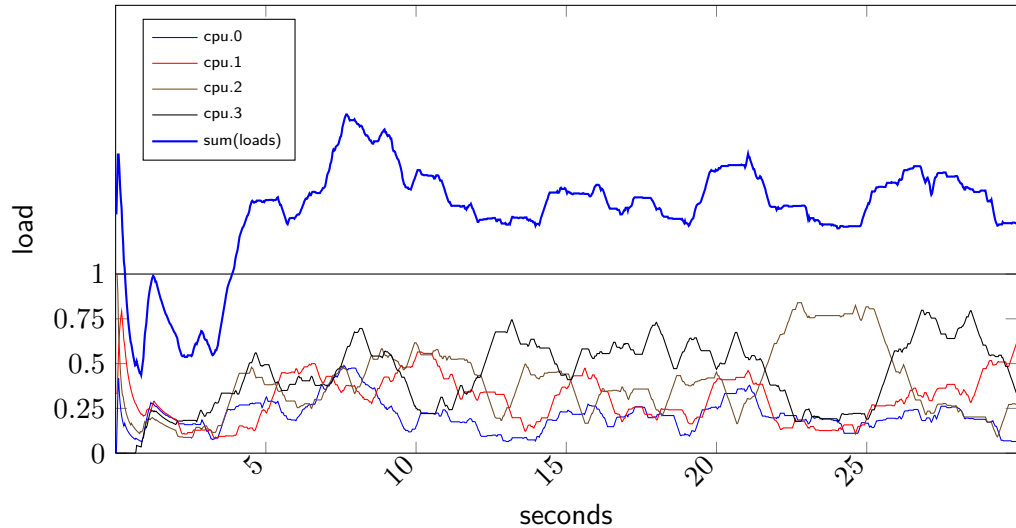
challenge, find the right frequency for *all* cores



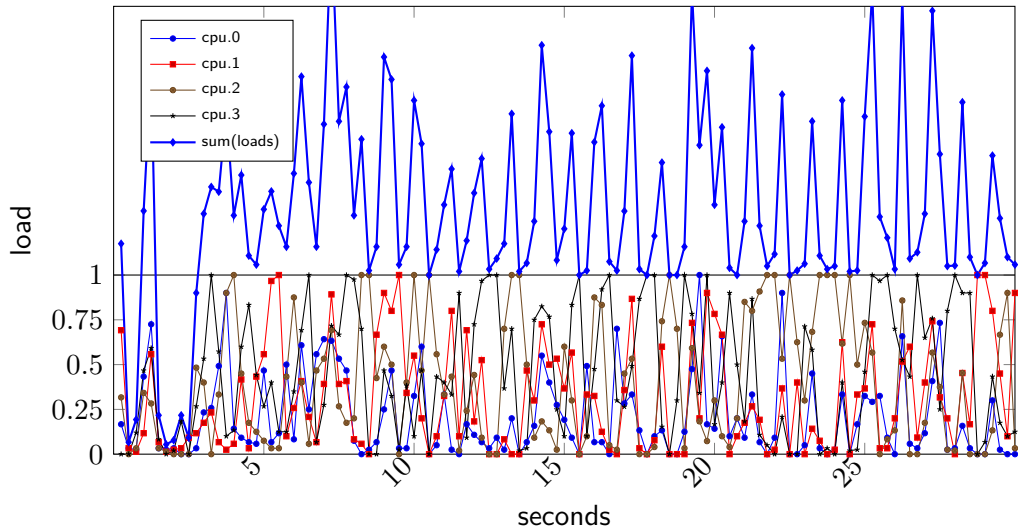
```
/*  
 * This function returns summary load of all CPUs. It was made so  
 * intentionally to not reduce performance in scenarios when several  
 * threads are processing requests as a pipeline -- running one at  
 * a time on different CPUs and waiting for each other.  
 */  
static int  
read_usage_times(int *load)
```



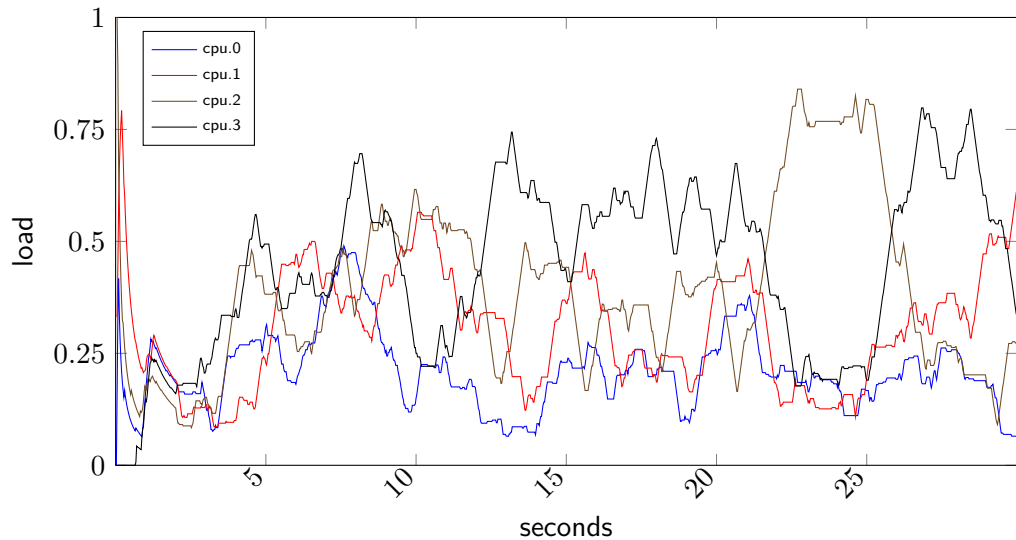
using the sum



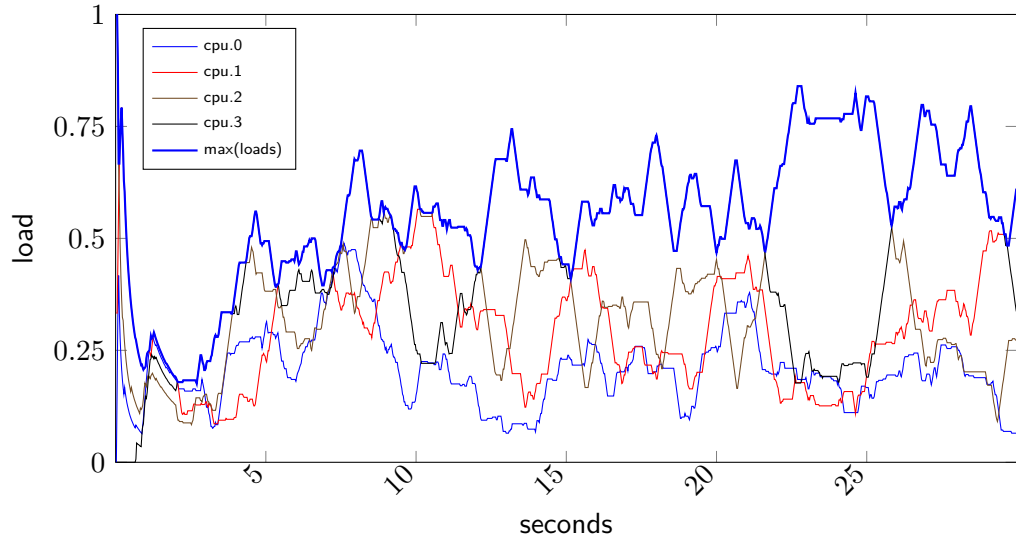
powerd flavoured



proposal



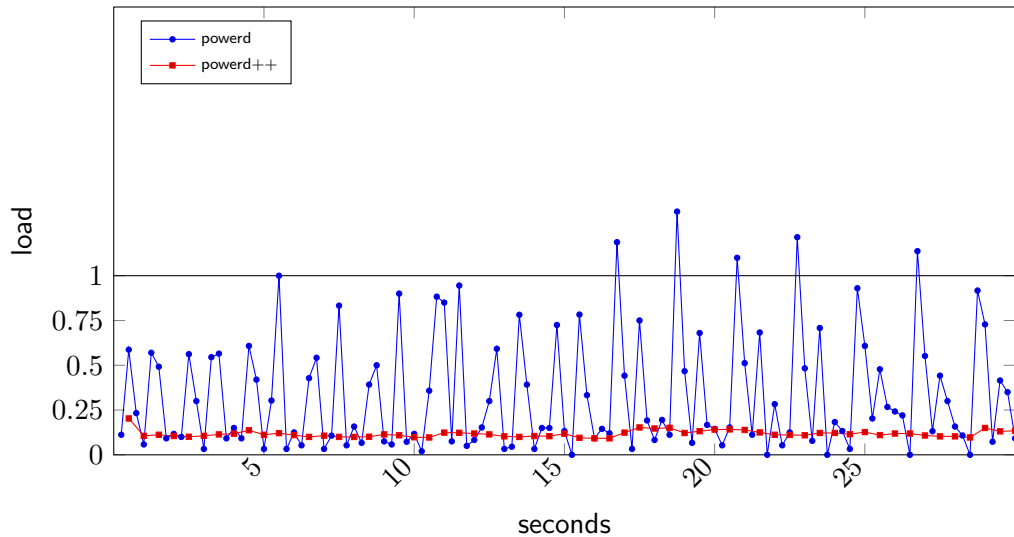
use the max



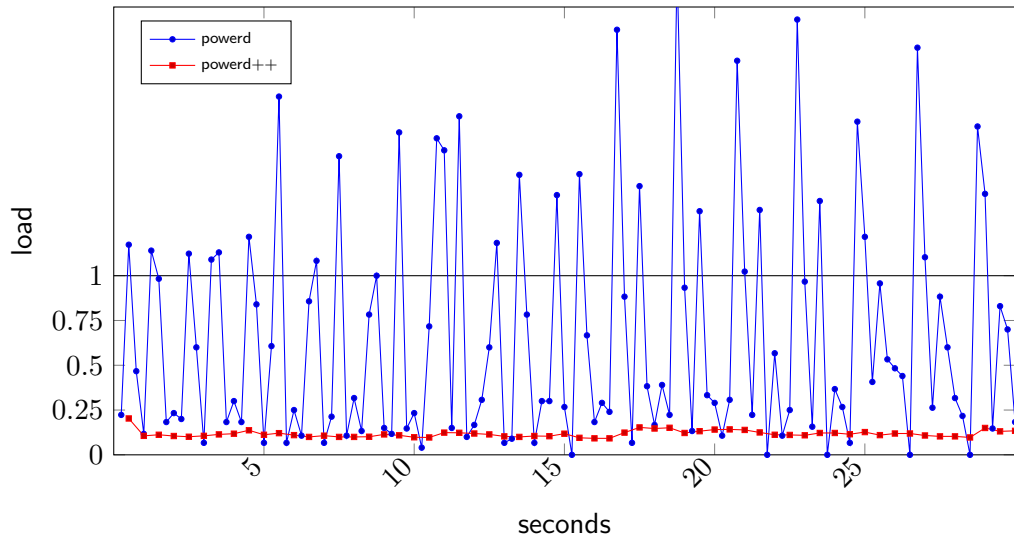
The Nail in the Coffin



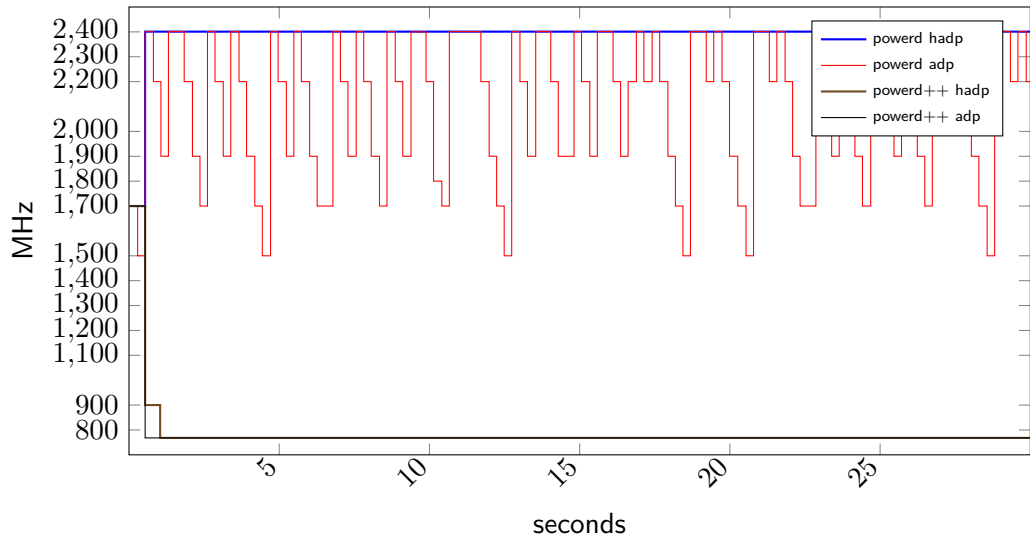
2 cores + HT



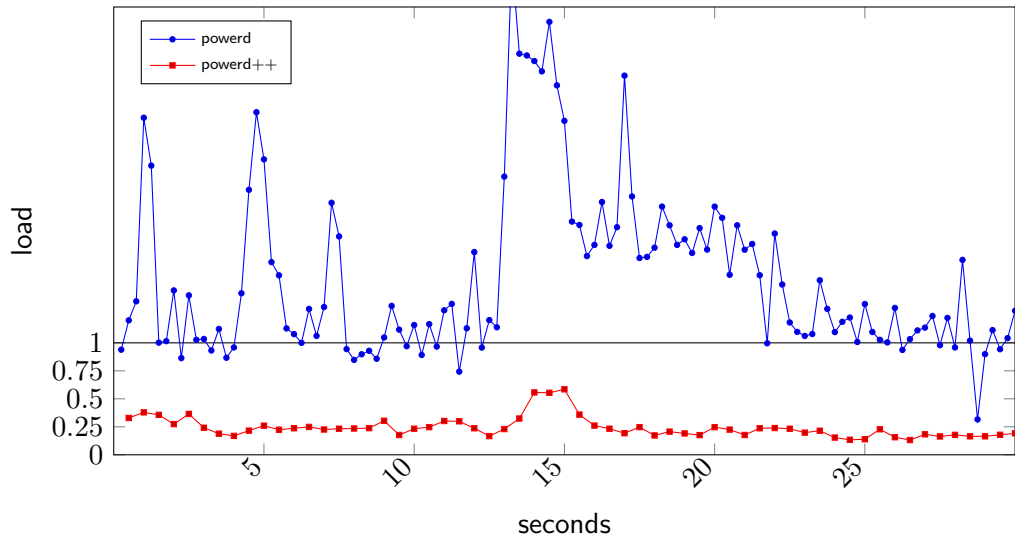
4 cores + HT



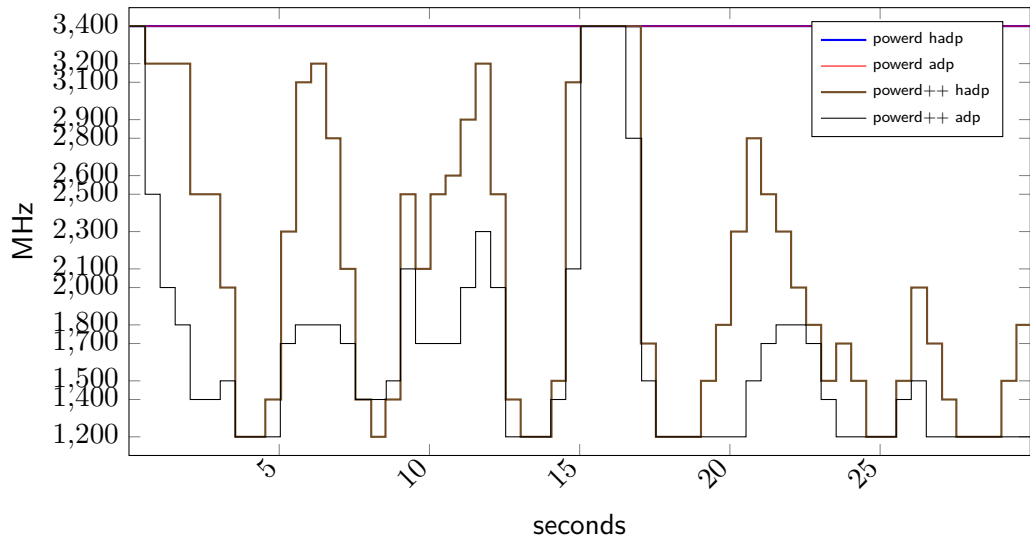
4 cores + HT



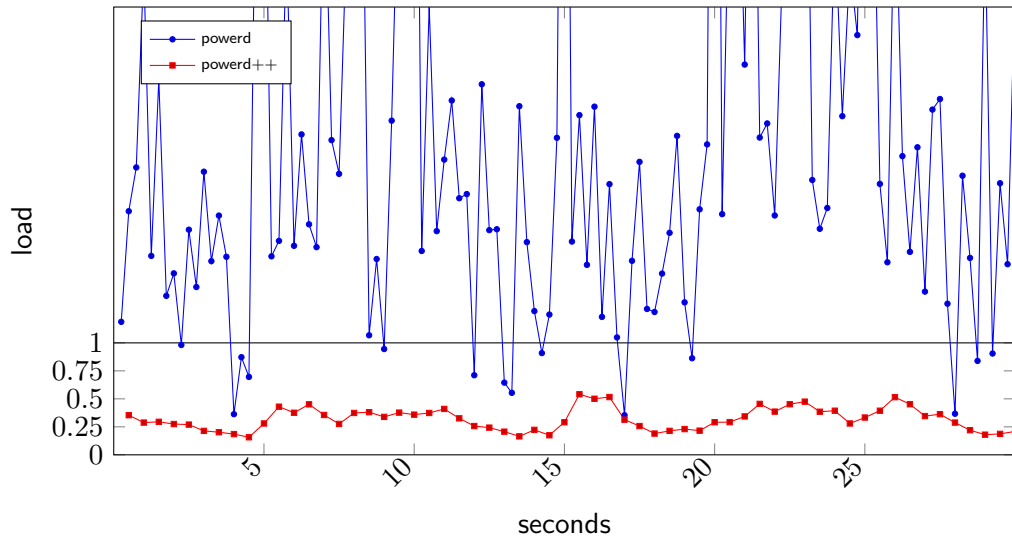
Venus (ncpu=24) Xeon E5-2643 v3



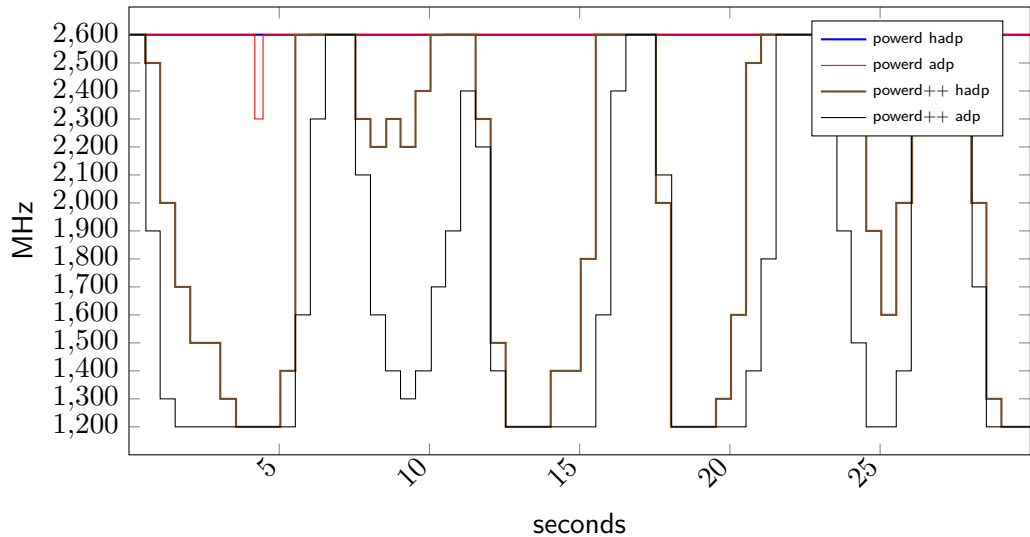
Venus (ncpu=24) Xeon E5-2643 v3



Ladon (ncpu=24) Xeon E5-2630



Ladon (ncpu=24) Xeon E5540



Per Core Clock Frequencies

demo



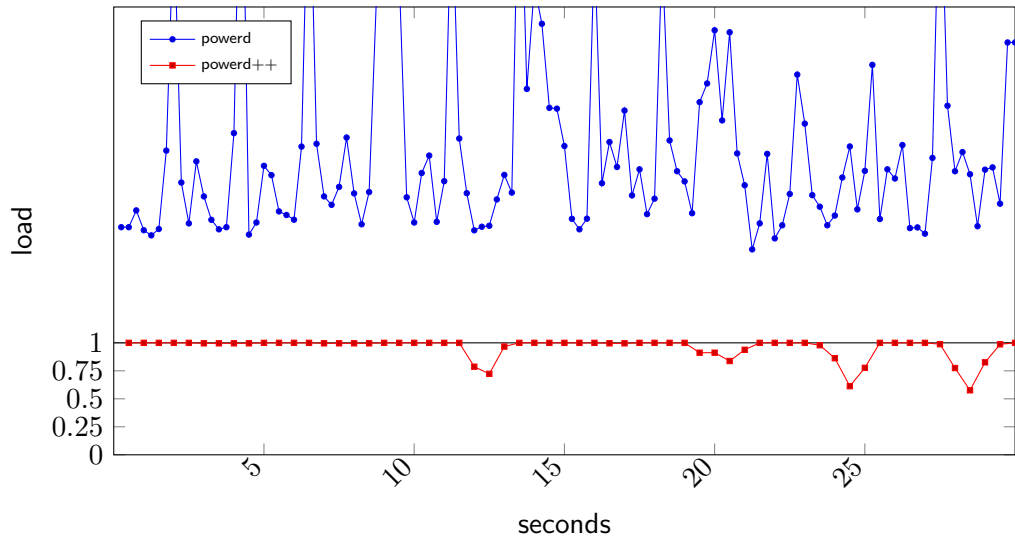
\(-)/

Praise the sun!

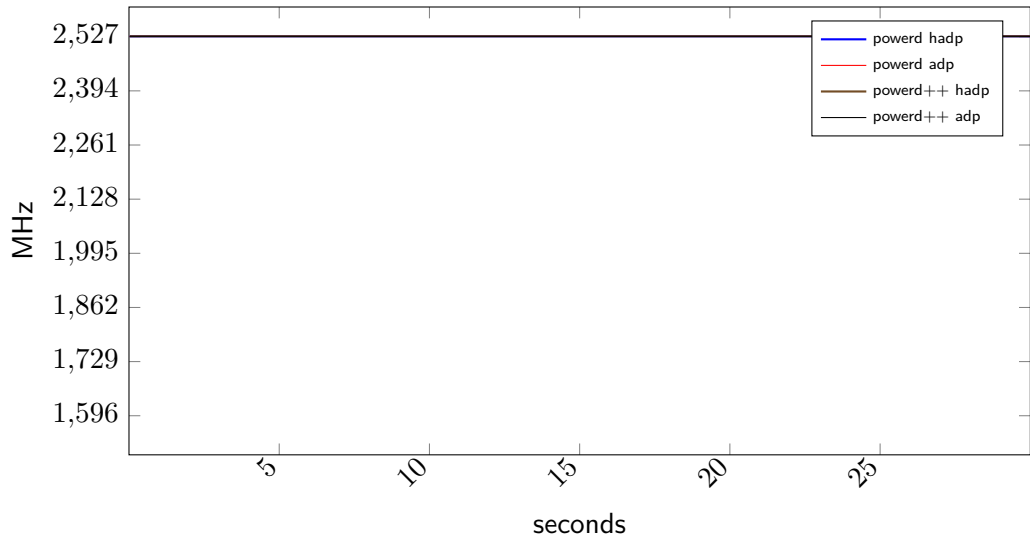
<https://github.com/lonkamikaze/powerdxx>



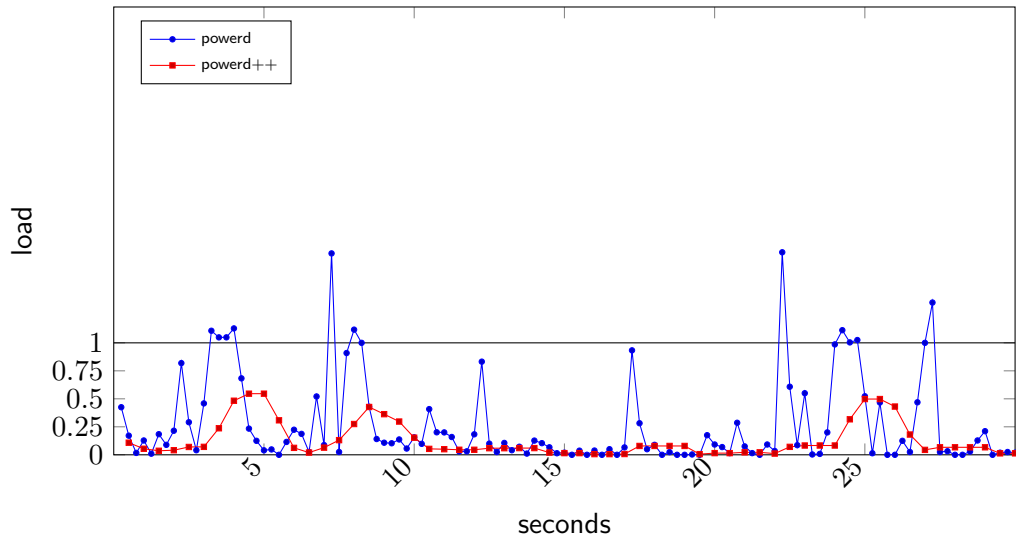
Inge (ncpu=16) Xeon E5540



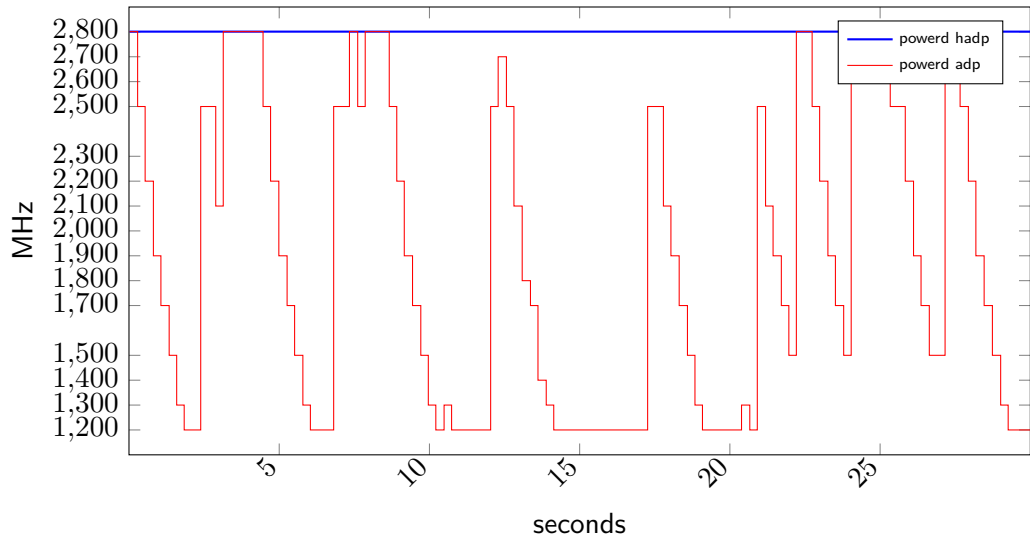
Inge (ncpu=16) Xeon E5540



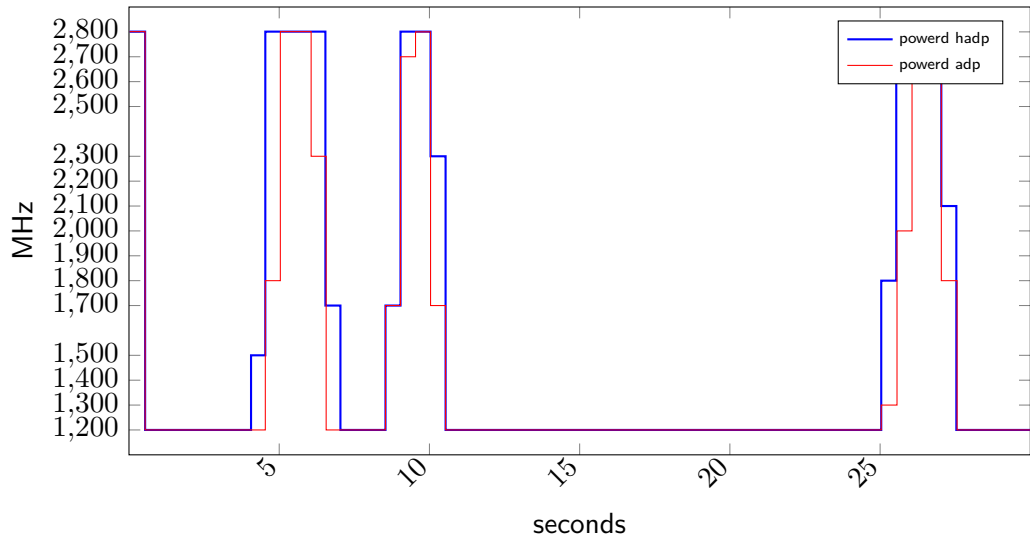
MCS3 (ncpu=40) Xeon E5-2680



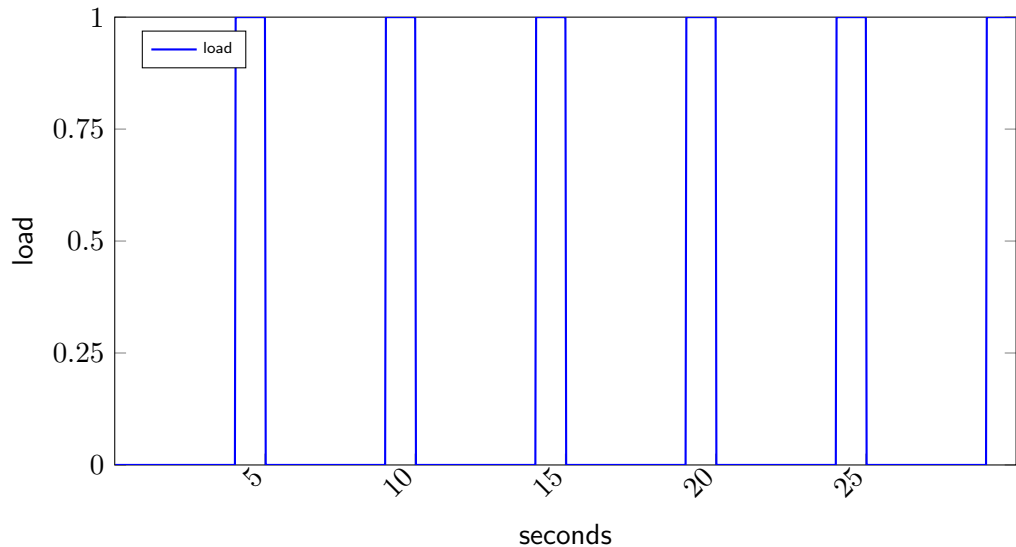
MCS3 (ncpu=40) Xeon E5-2680, powerd



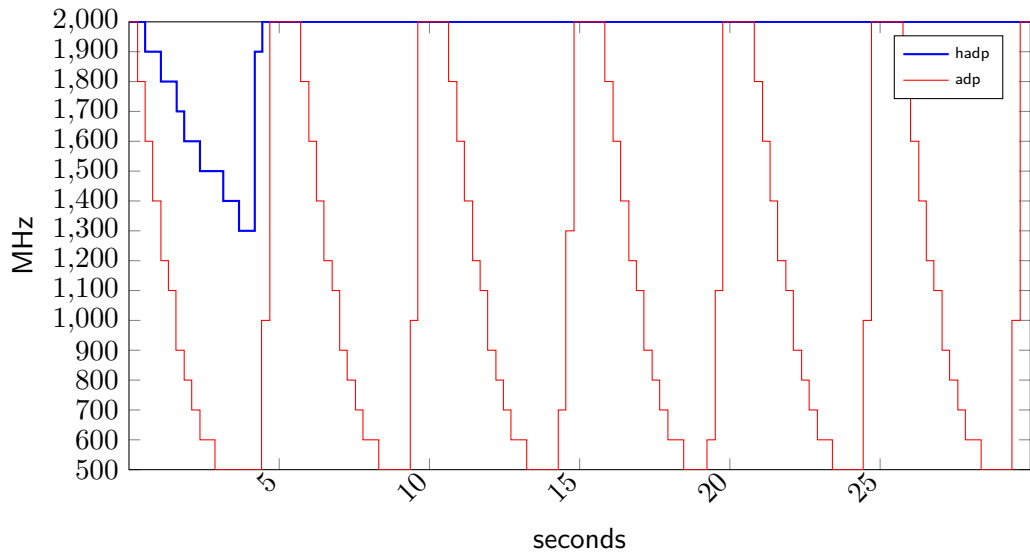
MCS3 (ncpu=40) Xeon E5-2680, powerd++



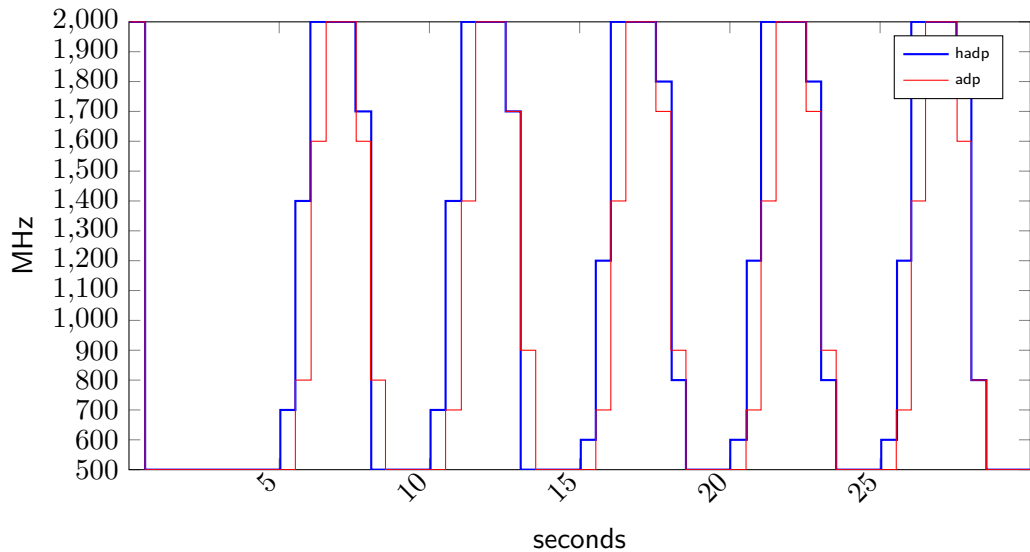
an artificial load



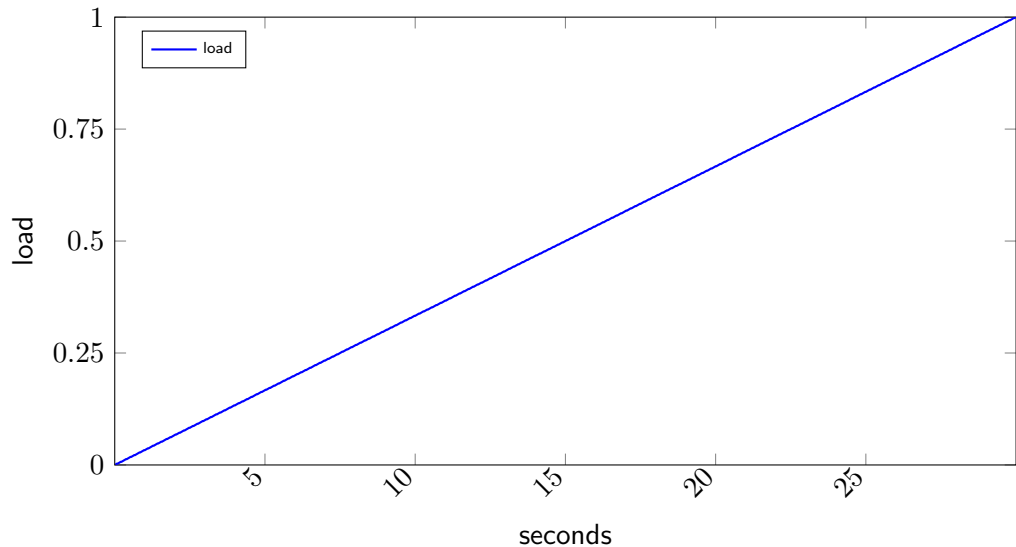
powerd



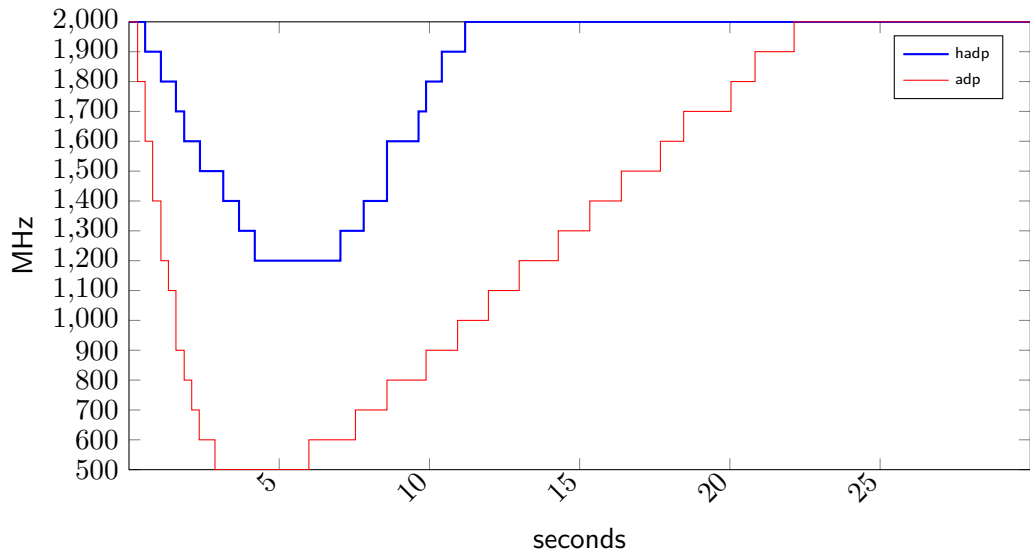
powerd++



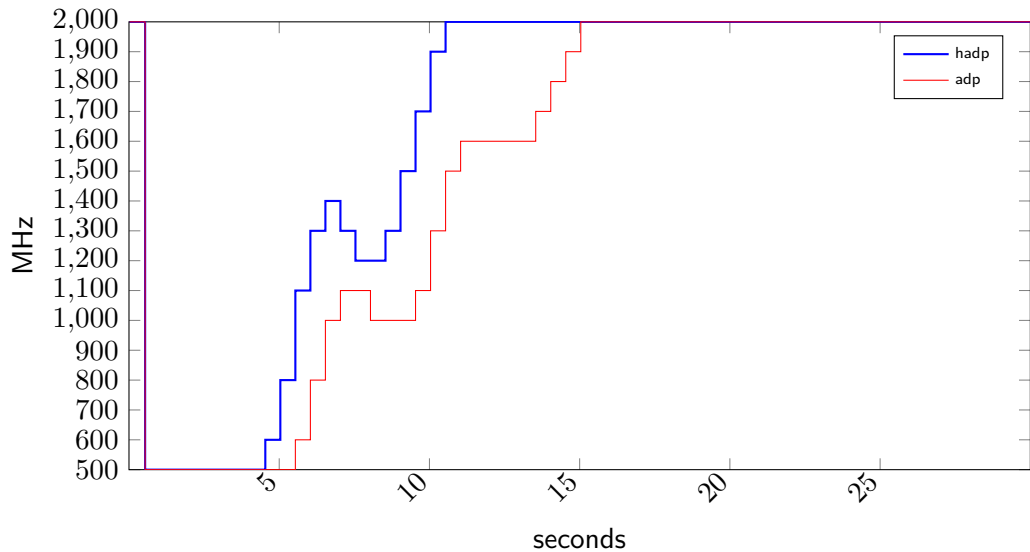
rampup



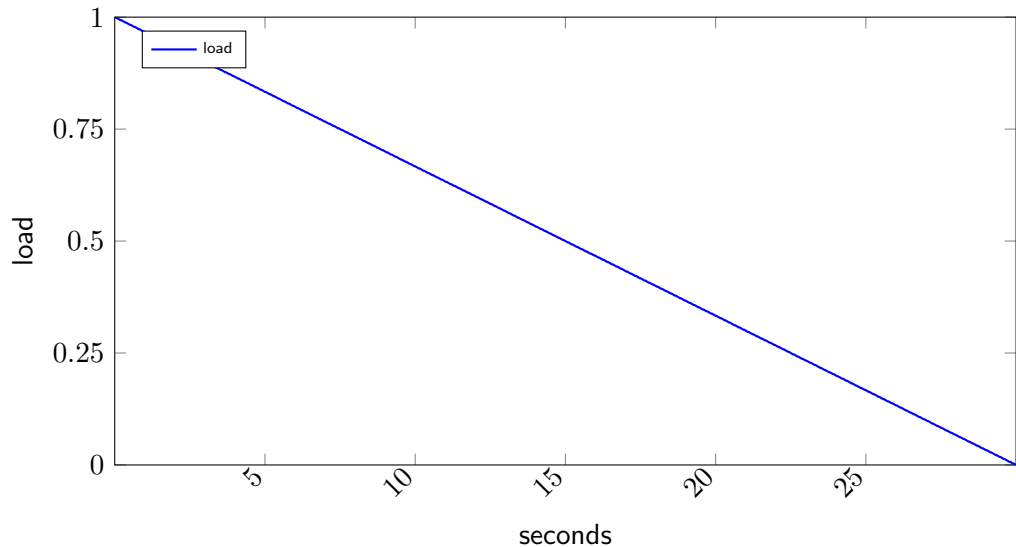
powerd



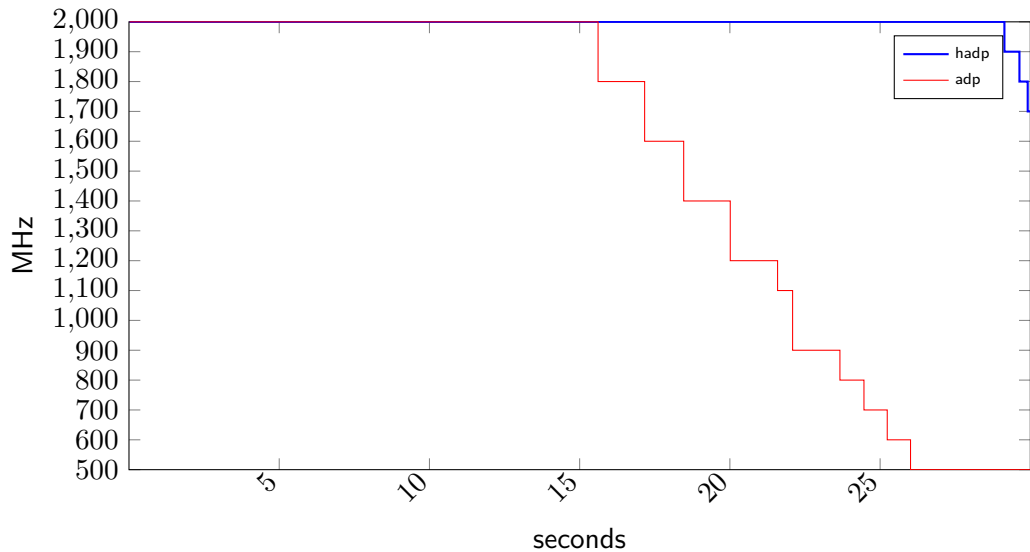
powerd++



rampdown



powerd



power++

