Proposing a Replacement for FreeBSD's powerd (Preview)

Or, how I tamed the fan of my notebook

Dominic Fandrey

Von Leitner-Institut für verteiltes Echtzeit-Java

GPN16, May 2016

Contents

Who?

Contents

Who?

What?

Contents

Who?

What?

Why?

Contents

Who?

What?

Why?

Challenges!

Contents

Who?

What?

Why?

Challenges!

How?

Contents

Who?

What?

Why?

Challenges!

How?

Conclusions!

kamikaze a.k.a. Dominic Fandrey

- Dominic Fandrey <kami@freebsd.org>
- M.Sc. (Computer Science)

kamikaze a.k.a. Dominic Fandrey

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

kamikaze a.k.a. Dominic Fandrey

- Dominic Fandrey <kami@freebsd.org>
- M.Sc. (Computer Science)
- Located in Europe/Karlsruhe
- Working as a researcher at an undisclosed polytechnic university
-

kamikaze a.k.a. Dominic Fandrey

- Dominic Fandrey <kami@freebsd.org>
- M.Sc. (Computer Science)
- Located in Europe/Karlsruhe
- Working as a researcher at an undisclosed polytechnic university
-

kamikaze a.k.a. Dominic Fandrey

- Dominic Fandrey <kami@freebsd.org>
- M.Sc. (Computer Science)
- Located in Europe/Karlsruhe
- Working as a researcher at an undisclosed polytechnic university
-

kamikaze a.k.a. Dominic Fandrey

- Dominic Fandrey <kami@freebsd.org>
- M.Sc. (Computer Science)
- Located in Europe/Karlsruhe
- Working as a researcher at an undisclosed polytechnic university
-

Definitions

- Load:

Definitions

- Load:
  - The fraction of CPU cycles not spent idle

Definitions

- Load:
  - The fraction of CPU cycles not spent idle
- P-State:

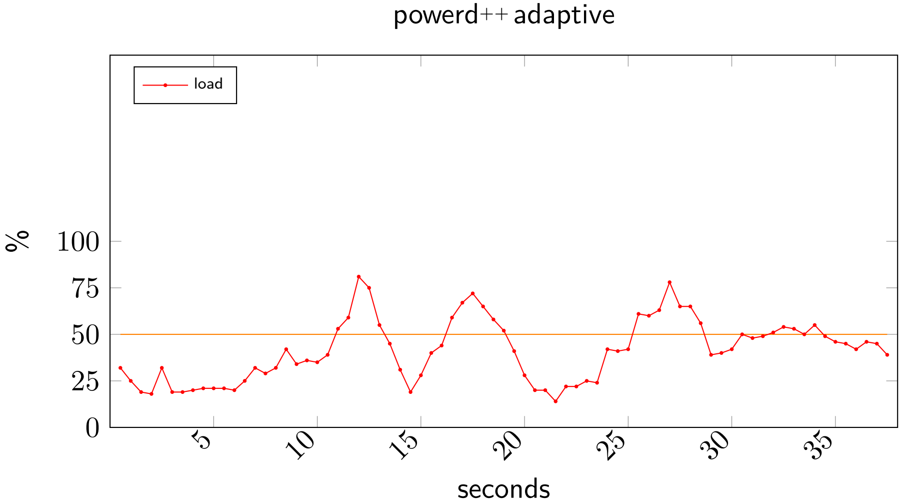
Definitions

- Load:
  - The fraction of CPU cycles not spent idle
- P-State:
  - Performance State, also frequently called stepping

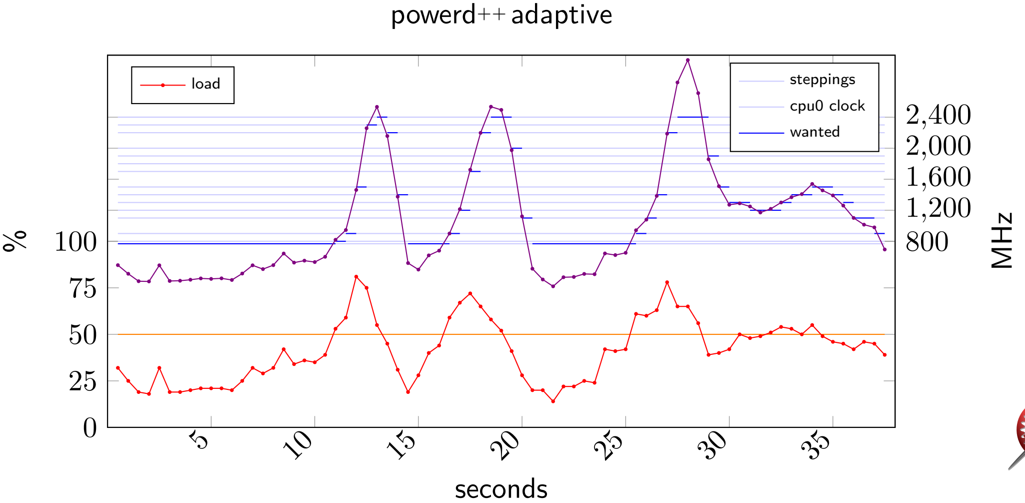
Definitions

- Load:
  - The fraction of CPU cycles not spent idle
- P-State:
  - Performance State, also frequently called stepping
  - A CPU mode of operation with a specific clock frequency and core voltage

CPU p-state control

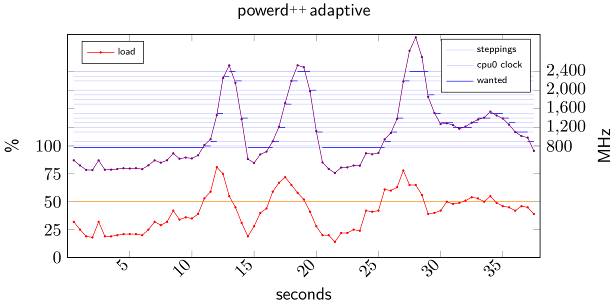


CPU p-state control



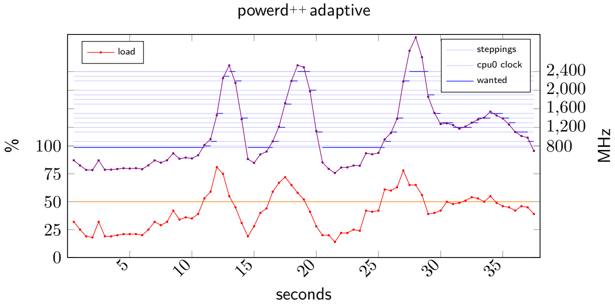
Why control p-state?

- Fan noise



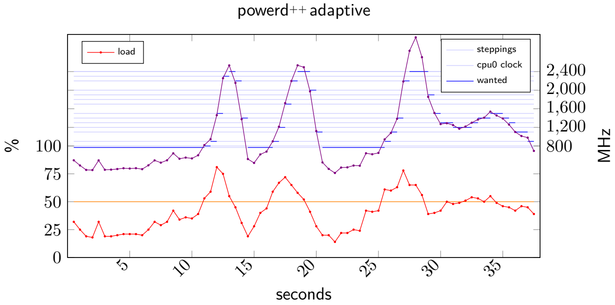
Why control p-state?

- Fan noise
- Battery/Energy conservation

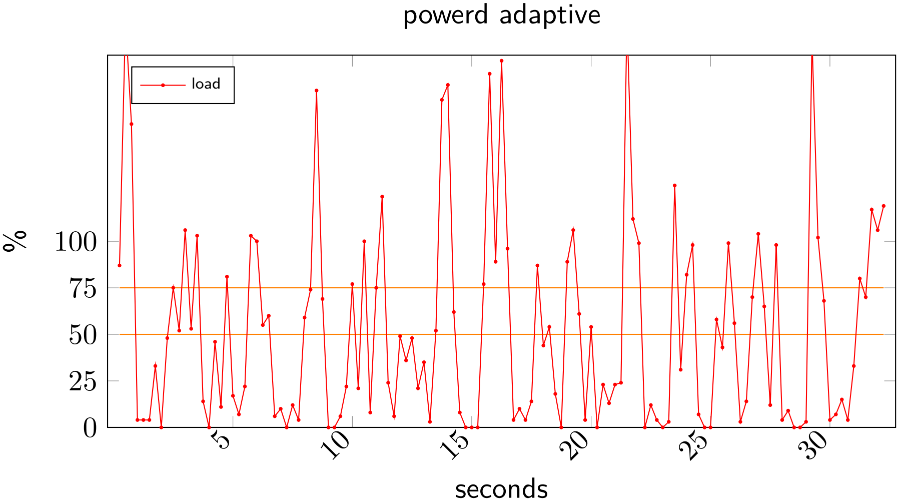


Why control p-state?

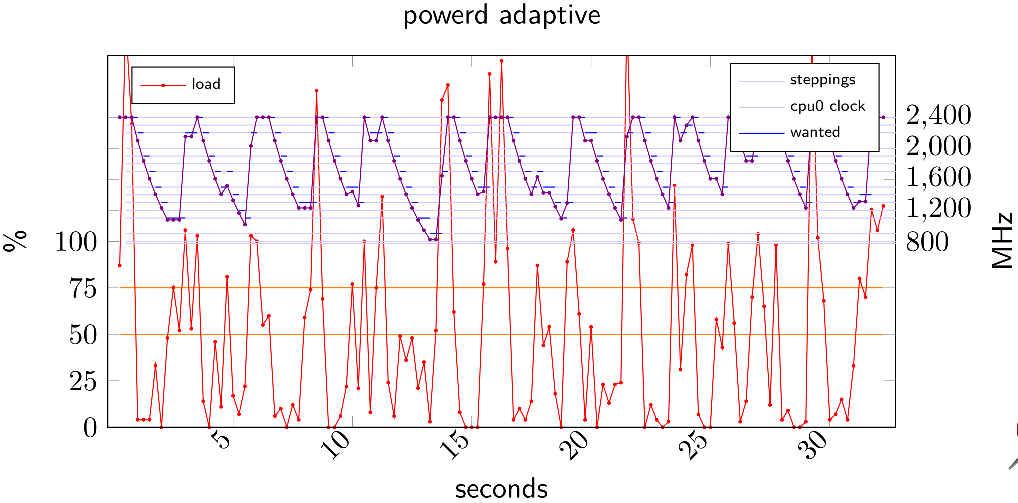
- Fan noise
- Battery/Energy conservation
- Hardware lifetime



Why replace powerd?

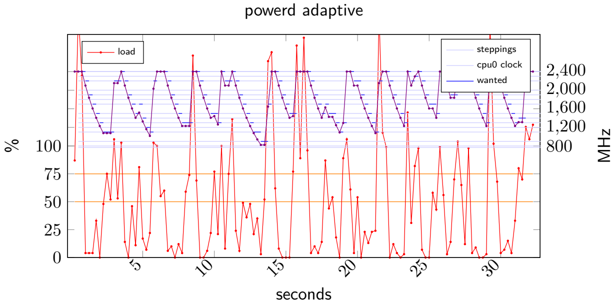


Why replace powerd?



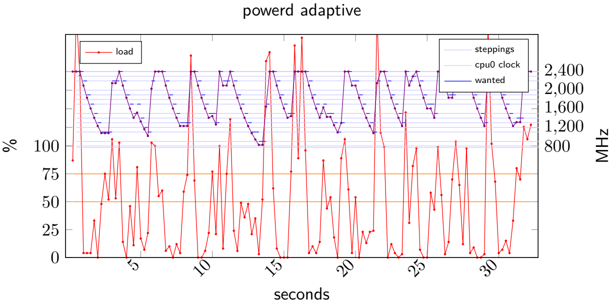
Why replace powerd?

- Broken load estimation



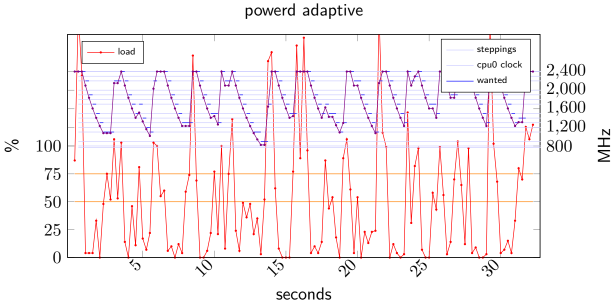
Why replace powerd?

- Broken load estimation
- Aggressive speeding



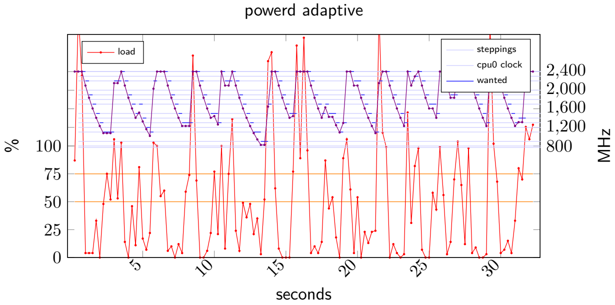
Why replace powerd?

- Broken load estimation
- Aggressive speeding
- Reluctant braking

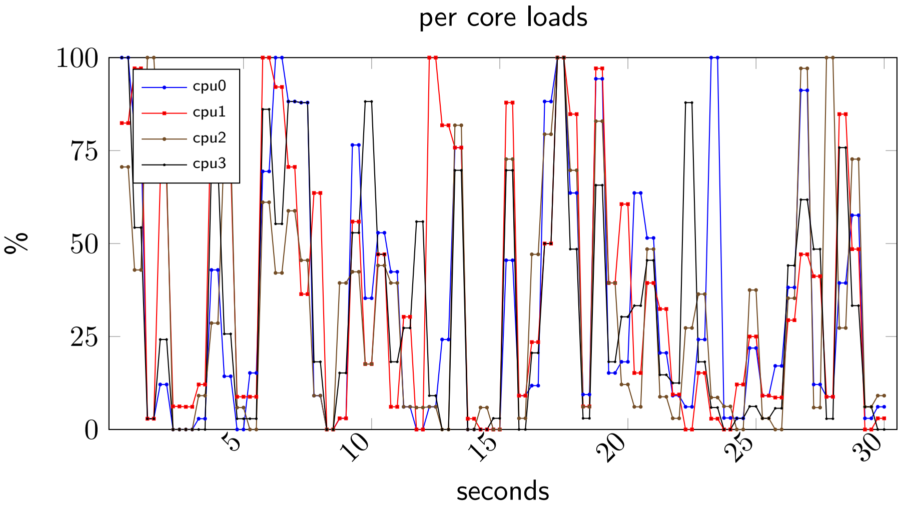


Why replace powerd?

- Broken load estimation
- Aggressive speeding
- Reluctant braking
- Excessive fan noise

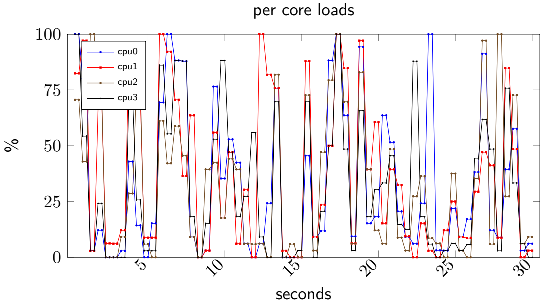


Measuring loads



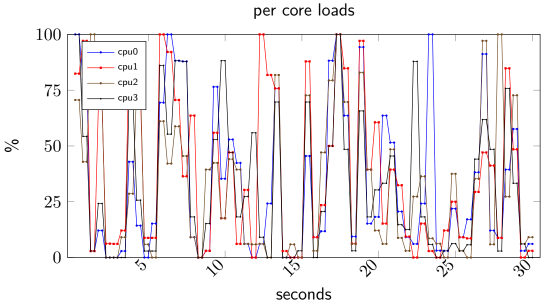
Measuring loads

- Load is noisy



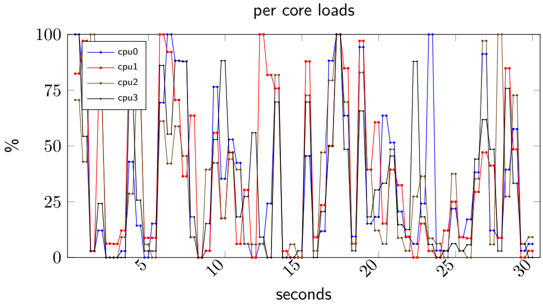
Measuring loads

- Load is noisy
- Load shifts



Measuring loads

- Load is noisy
- Load shifts
- Load saturates



Control algorithm

- Contradicting goals:

Control algorithm

- Contradicting goals:
  - System should be responsive

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient
- P-States:

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient
- P-States:
  - P-States lie

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient
- P-States:
  - P-States lie
  - Extremely low nonsense p-states

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient
- P-States:
  - P-States lie
  - Extremely low nonsense p-states
  - P-State switching has a cost

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient
- P-States:
  - P-States lie
  - Extremely low nonsense p-states
  - P-State switching has a cost

- Conclusions:

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient
- P-States:
  - P-States lie
  - Extremely low nonsense p-states
  - P-State switching has a cost

- Conclusions:
  - Make the right compromises

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient
- P-States:
  - P-States lie
  - Extremely low nonsense p-states
  - P-State switching has a cost

- Conclusions:
  - Make the right compromises
  - Make it tunable

Control algorithm

- Contradicting goals:
  - System should be responsive
  - System should be energy efficient
- P-States:
  - P-States lie
  - Extremely low nonsense p-states
  - P-State switching has a cost

- Conclusions:
  - Make the right compromises
  - Make it tunable
  - Set sane defaults

Summary

powerd++ powerd

versus

Summary

powerd++

- Any granularity p-state changes

versus

powerd

Summary

powerd++

- Any granularity p-state changes

powerd

- Global p-state changes only

versus

Summary

powerd++

- Any granularity p-state changes
- Maximum load

powerd

- Global p-state changes only

versus

Summary

powerd++

- Any granularity p-state changes
- Maximum load

powerd

- Global p-state changes only
- Sum of loads

versus

Summary

powerd++

- Any granularity p-state changes
- Maximum load
- Aim for a load target

versus

powerd

- Global p-state changes only
- Sum of loads

Summary

powerd++

- Any granularity p-state changes
- Maximum load
- Aim for a load target

versus

powerd

- Global p-state changes only
- Sum of loads
- Hysteresis

Summary

powerd++

- Any granularity p-state changes
- Maximum load
- Aim for a load target
- Filter load to ignore short spikes

versus

powerd

- Global p-state changes only
- Sum of loads
- Hysteresis

Summary

powerd++

- Any granularity p-state changes
- Maximum load
- Aim for a load target
- Filter load to ignore short spikes

versus

powerd

- Global p-state changes only
- Sum of loads
- Hysteresis
- Aggressively tuned for responsiveness

Summary

powerd++

- Any granularity p-state changes
- Maximum load
- Aim for a load target
- Filter load to ignore short spikes
- Explicit CLA syntax like `--max 1.2ghz`

versus

powerd

- Global p-state changes only
- Sum of loads
- Hysteresis
- Aggressively tuned for responsiveness

Summary

powerd++

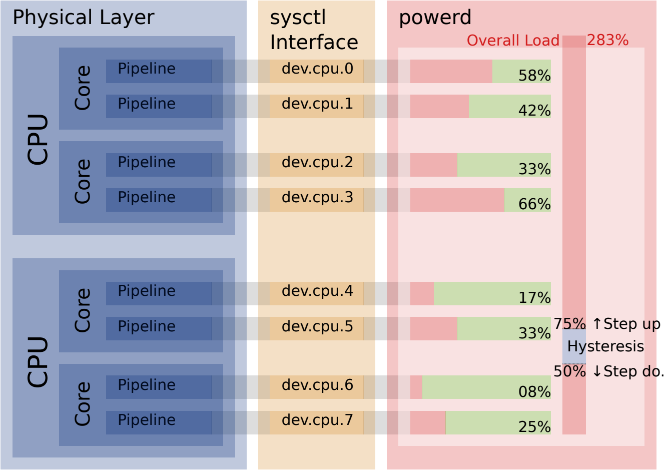
- Any granularity p-state changes
- Maximum load
- Aim for a load target
- Filter load to ignore short spikes
- Explicit CLA syntax like `--max 1.2ghz`

versus

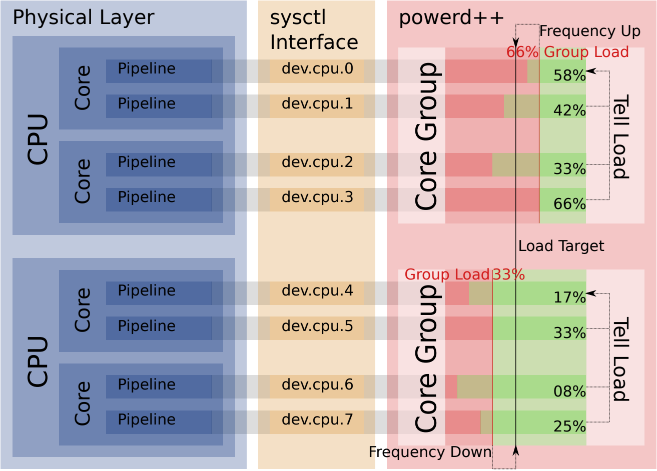
powerd

- Global p-state changes only
- Sum of loads
- Hysteresis
- Aggressively tuned for responsiveness
- Hard coded units `-M 1200`

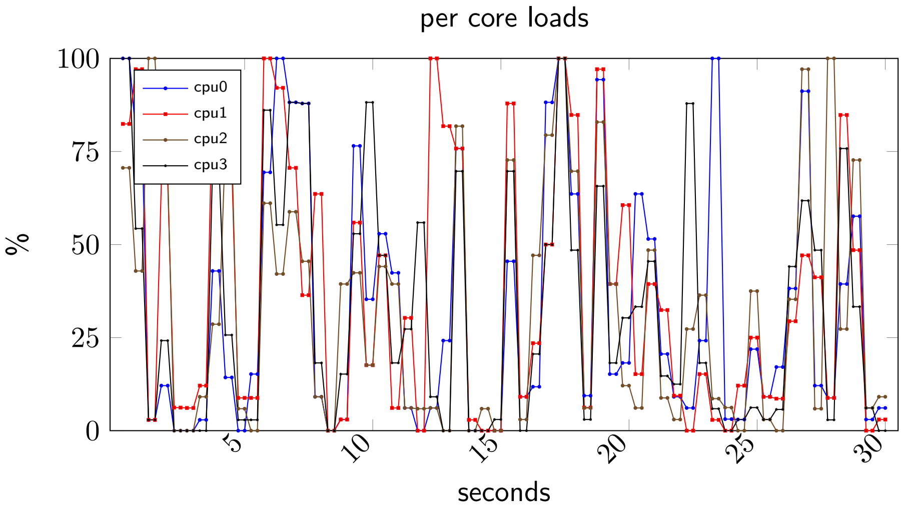
Control algorithm (powerd)



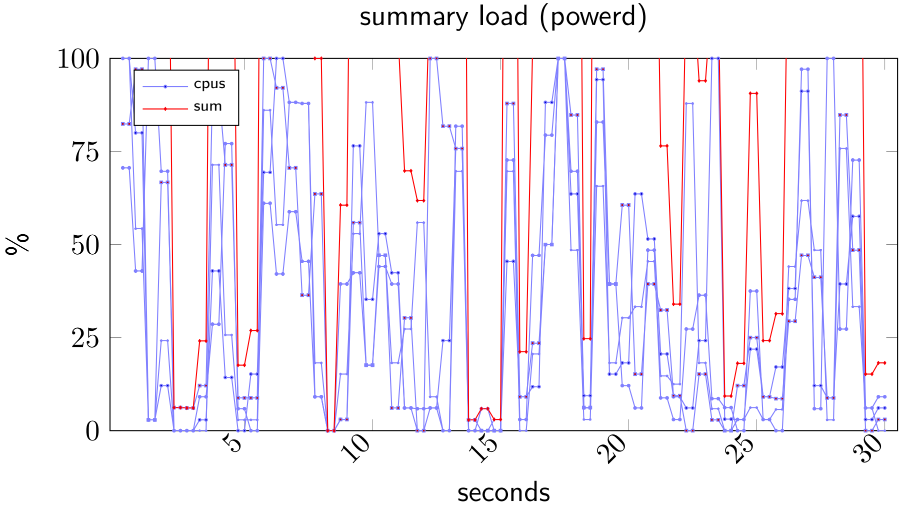
Control algorithm (powerd++)



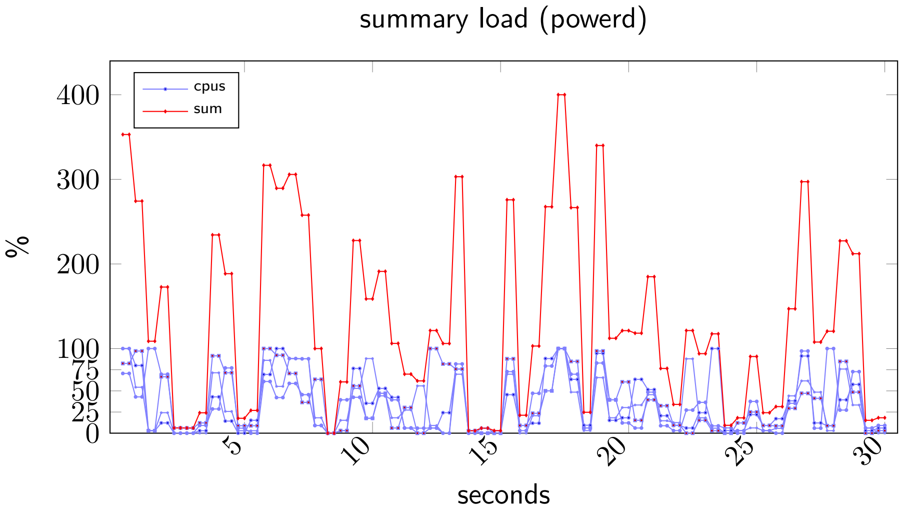
Dealing with signal noise



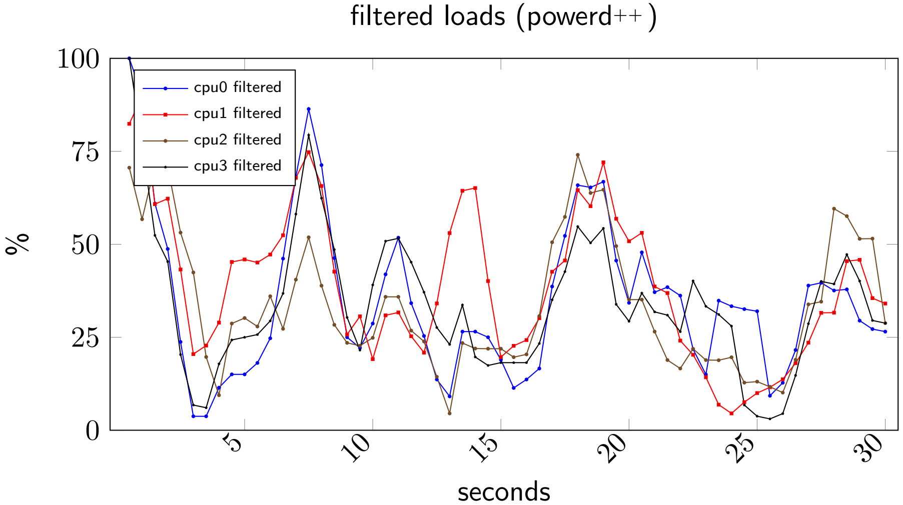
Dealing with signal noise



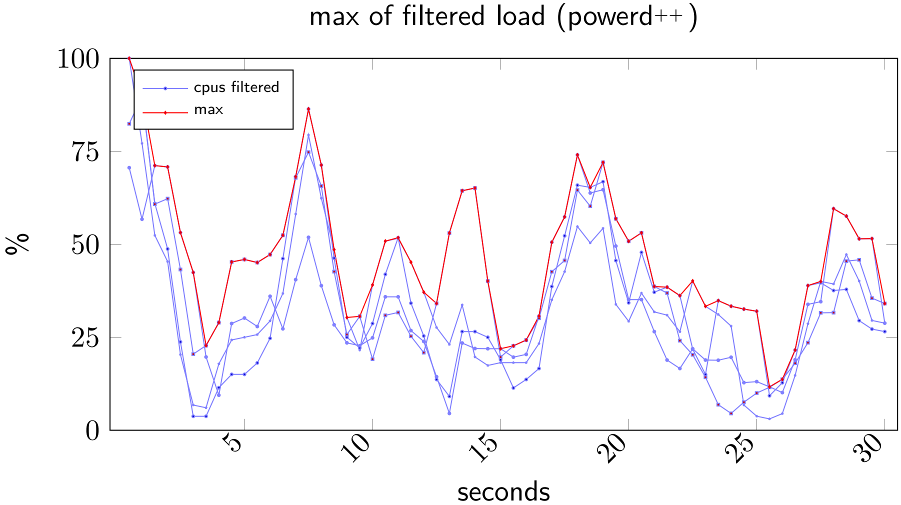
Dealing with signal noise



Dealing with signal noise



Dealing with signal noise



Solved problems

- Load signal noise

Solved problems

- Load signal noise
- Lower sample rate, gliding average

Solved problems

- Load signal noise
- Low multi core loads

- Lower sample rate, gliding average

Solved problems

- Load signal noise
- Low multi core loads

- Lower sample rate, gliding average
- Without breaking single core loads

Unsolved problems

- High frequency core hopping

Unsolved problems

- High frequency core hopping

- This is rare

Unsolved problems

- High frequency core hopping
- P-States that lie reduce accuracy

- This is rare

Unsolved problems

- High frequency core hopping
- P-States that lie reduce accuracy

- This is rare
- Ignoring this works well enough

Unsolved problems

- High frequency core hopping
- P-States that lie reduce accuracy
- `kern_clock.c` only supports global frequency changes

- This is rare
- Ignoring this works well enough

Unsolved problems

- High frequency core hopping
- P-States that lie reduce accuracy
- `kern_clock.c` only supports global frequency changes

- This is rare
- Ignoring this works well enough
- This is fixable, but may break scheduling

\(-)/

Praise the sun!

`https://github.com/lonkamikaze/powerdxx`