Energy-Centric Dynamic Fan Control

B. Pradelle, N. Triquenaux, J. C. Beyler, W. Jalby

Exascale computing research

UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES

Outline

- Introduction to fan control
- DFaCE
- Experimental results
- Conclusion

Introduction to fan control

- Fans are a good target for energy savings
  - Used in many computers (still)
    - Water cooling is coming
  - Consume a large part of energy
    - Fans are 3rd or 4th most consuming component

Introduction to fan control

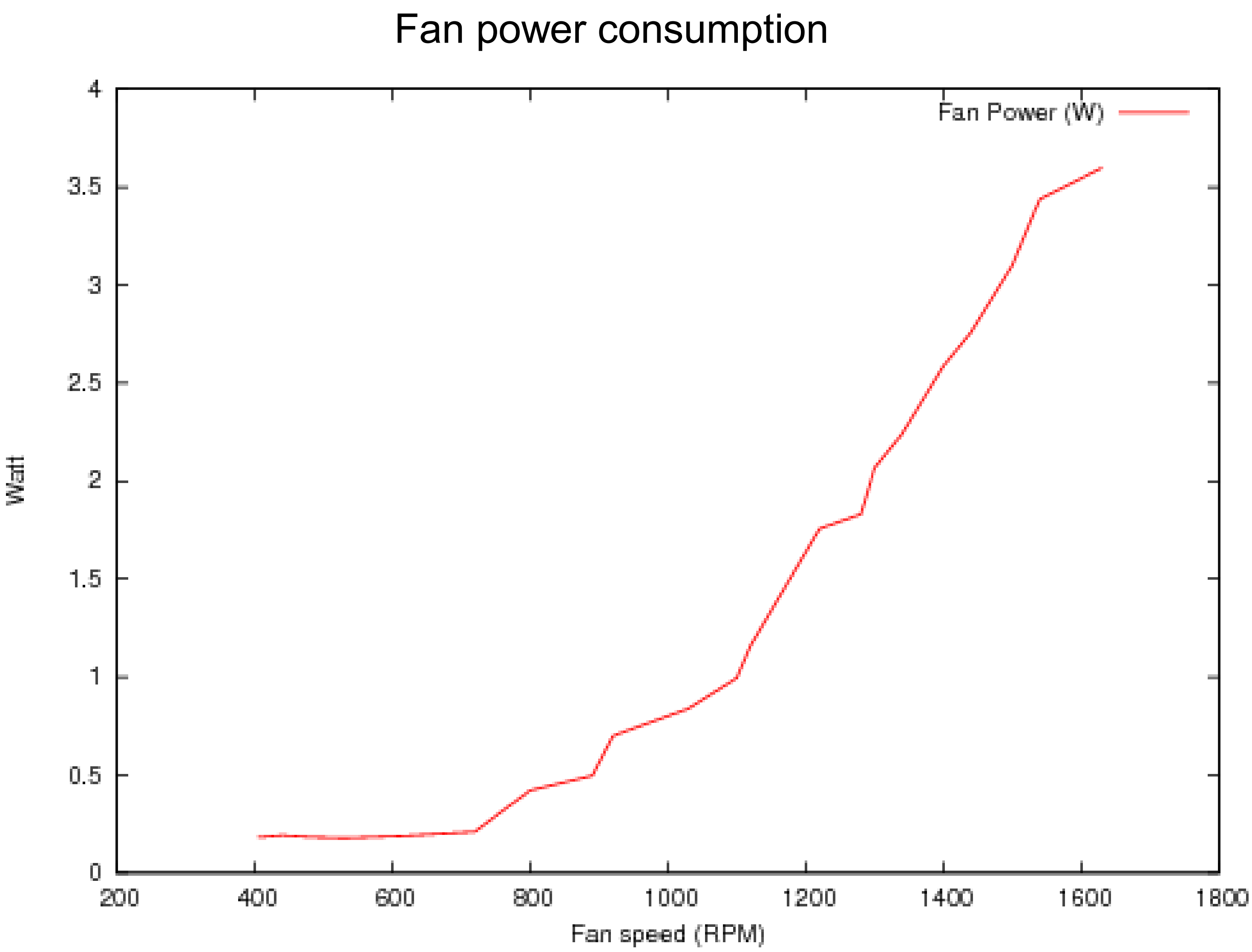
- Fans control is mostly thermal-directed
  - Uses a default temperature target
  - Only avoids hardware failure
  - Has no consideration for energy

Introduction to fan control

- High fan speeds
  - Large fan power consumption
- Low fan speeds
  - High CPU temperature
  - Large power leakage

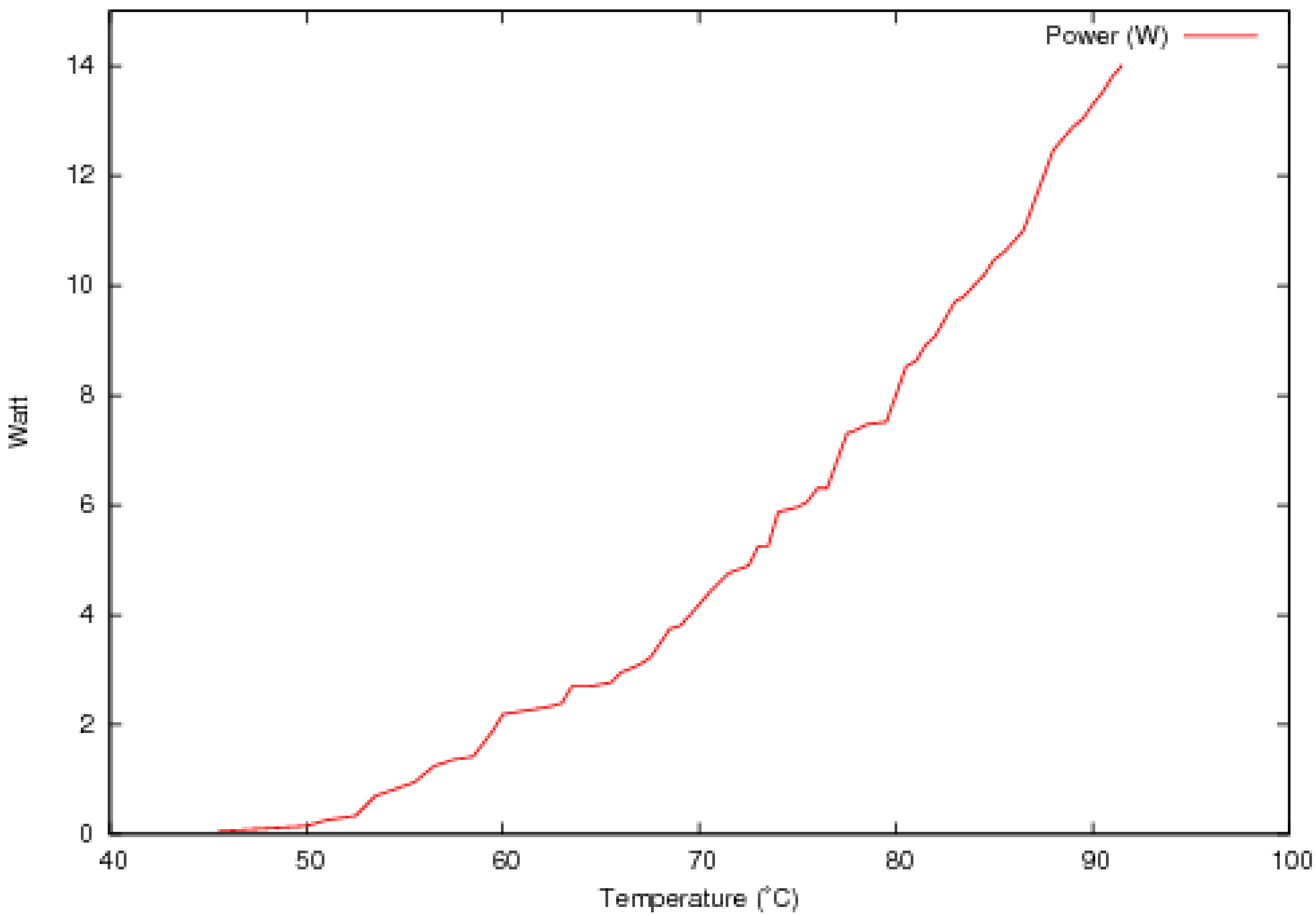
- Which fan speed minimizes energy consumption?

Introduction to fan control

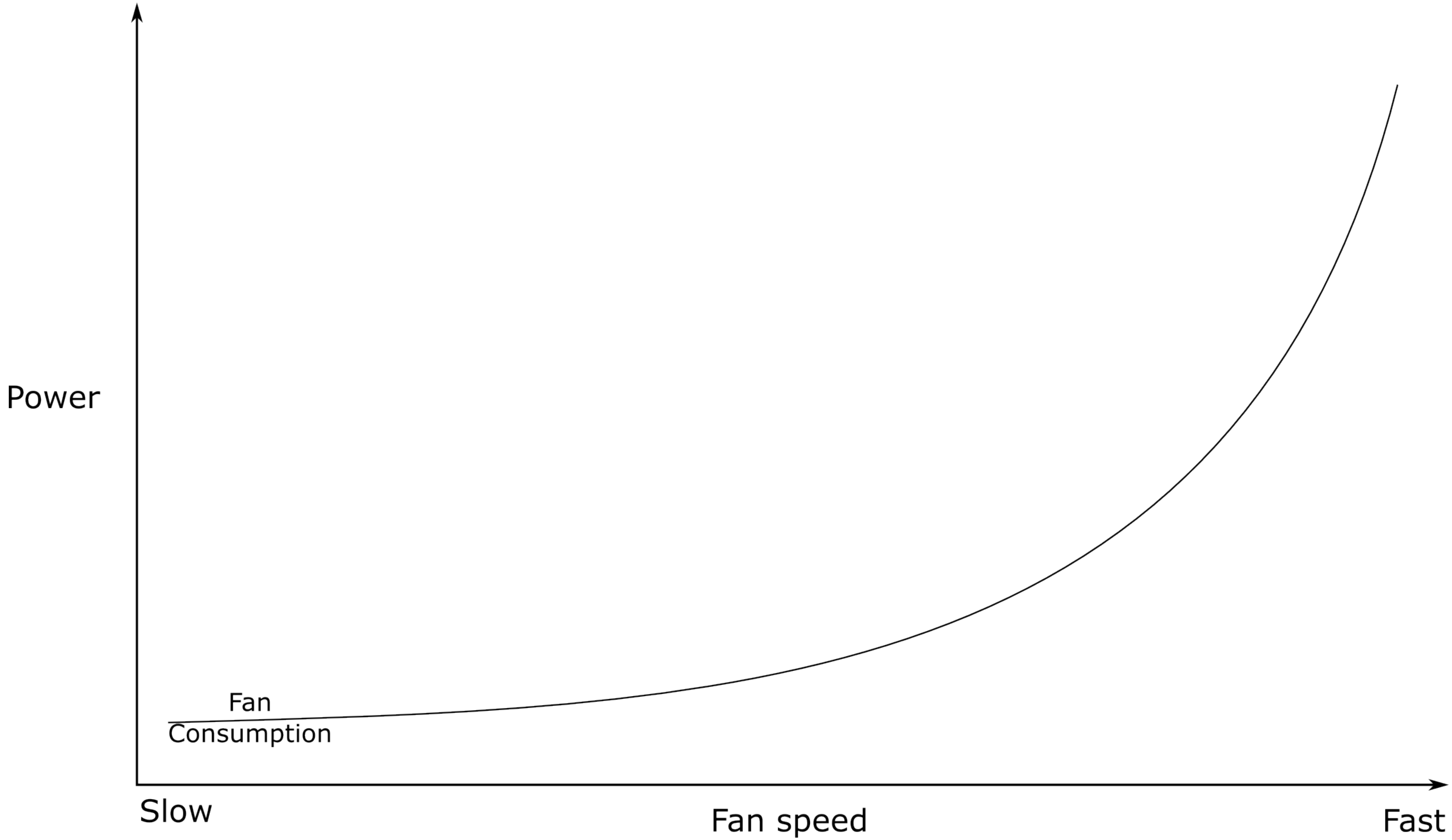


Introduction to fan control

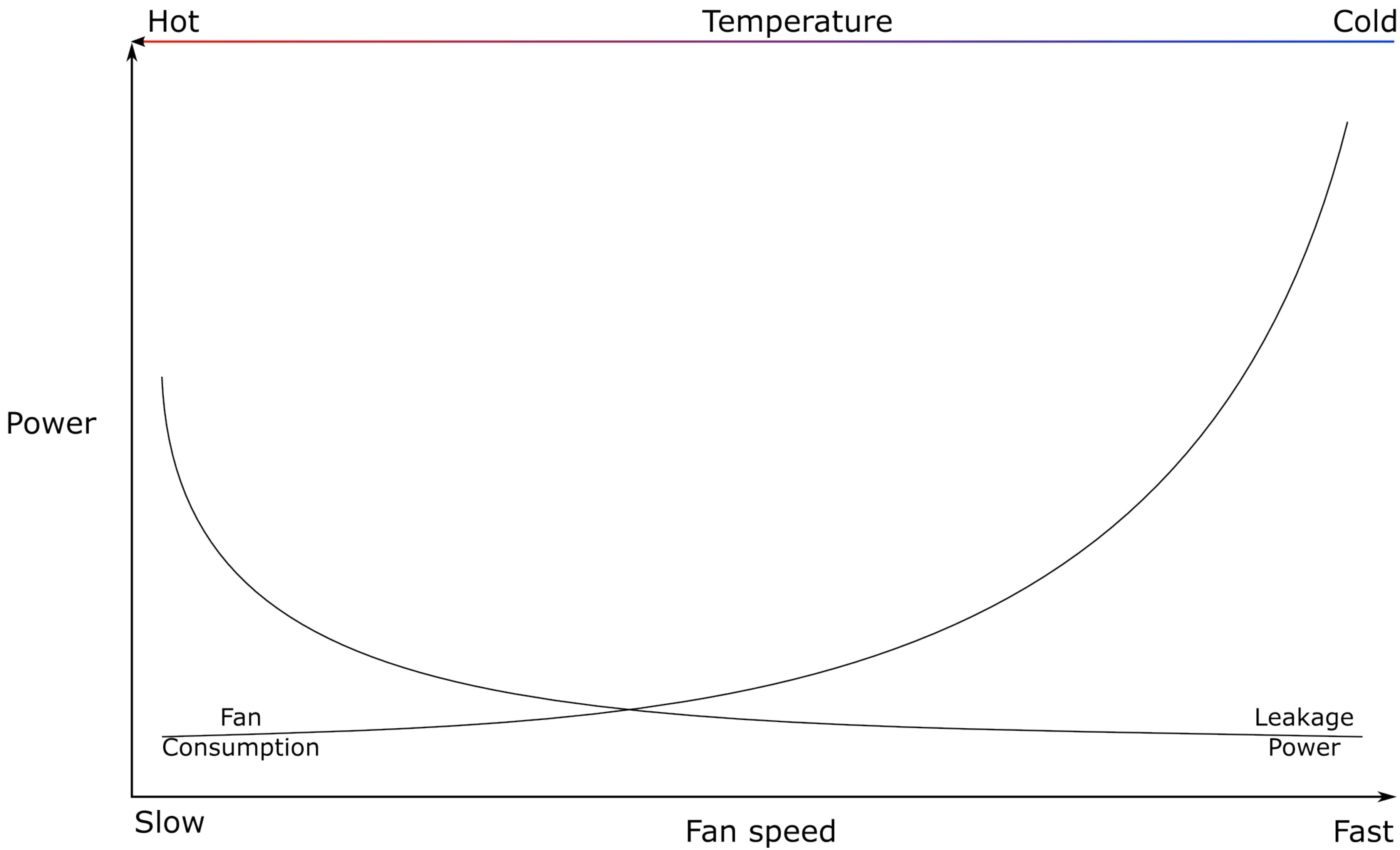
CPU leakage power



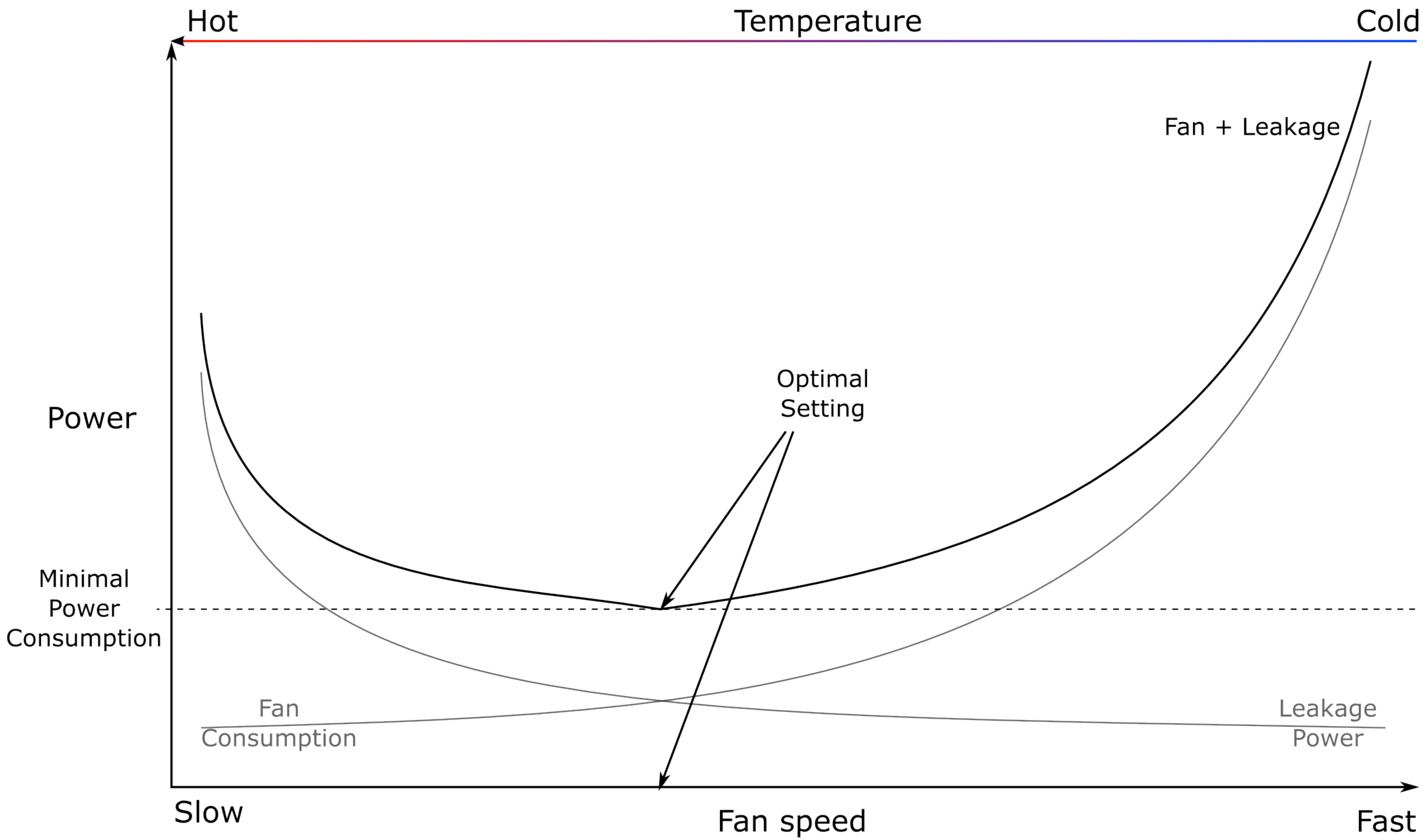
Introduction to fan control



Introduction to fan control



Introduction to fan control



Introduction to fan control

- Computers often have several fans
  - Each fan has 256 speed settings on Linux
  - 3 fans = 16,777,216 possible settings
- A fan setting is optimal for a specific workload
- A fast setting evaluation is required

DFaCE
(Dynamic Fan Controller for Energy)

DFaCE

- Objectives
  - Save energy
  - Avoid critical temperatures
  - Manage several fans
  - Keep low overhead

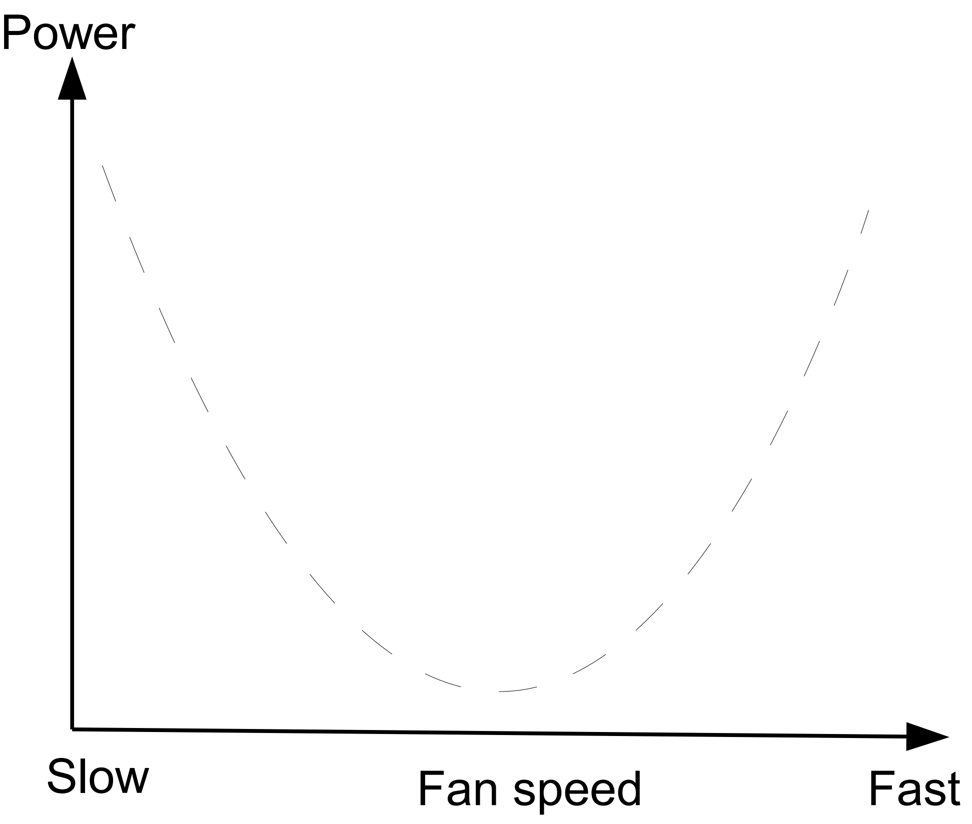
DFaCE

- Power consumption/fan setting is convex
  - Fans consume more energy when running faster
  - CPU consumes more energy when fans runs slower

- Hill-climbing optimization can be used

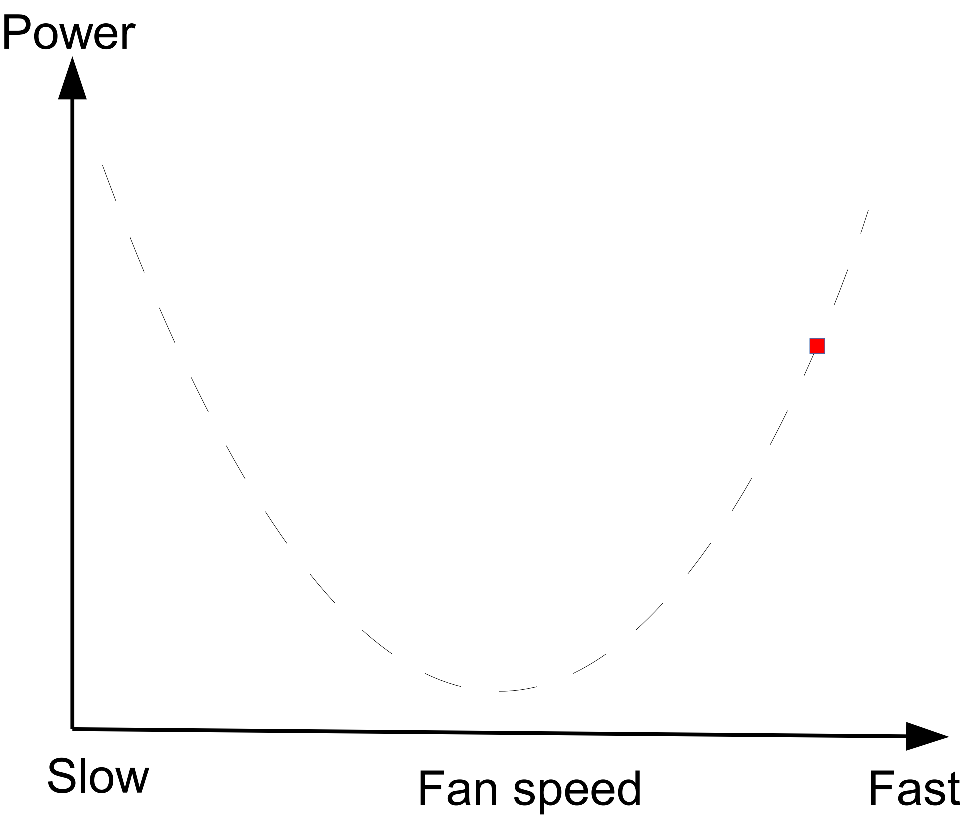
DFaCE

- Hill climbing:



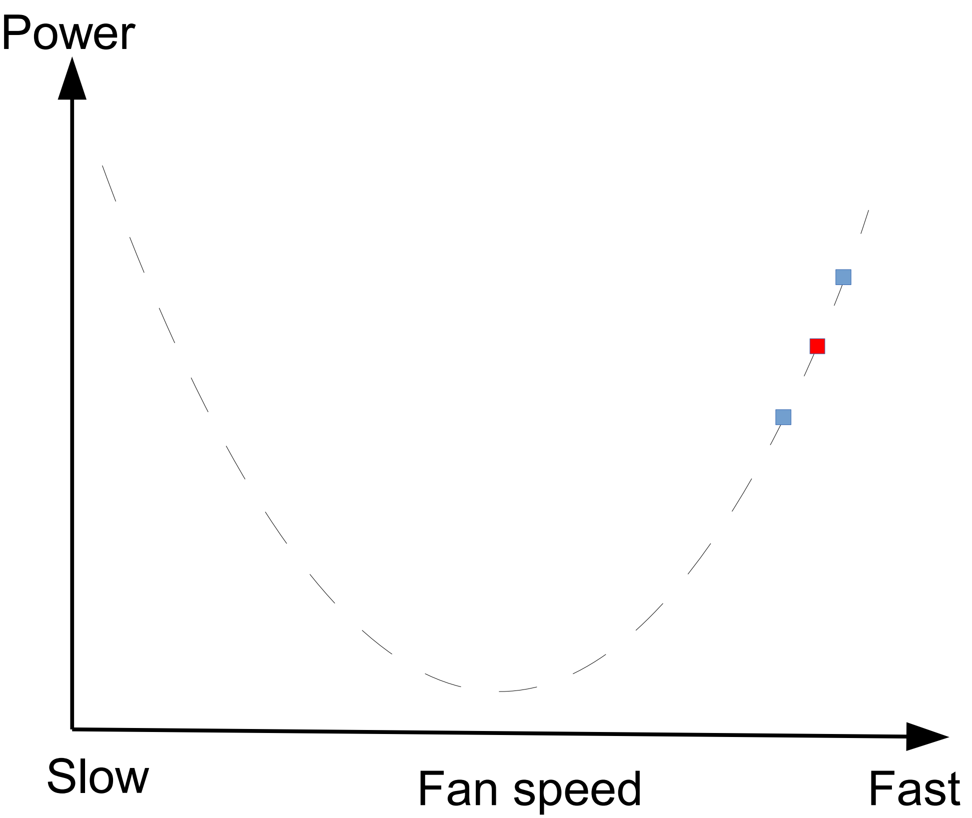
DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting



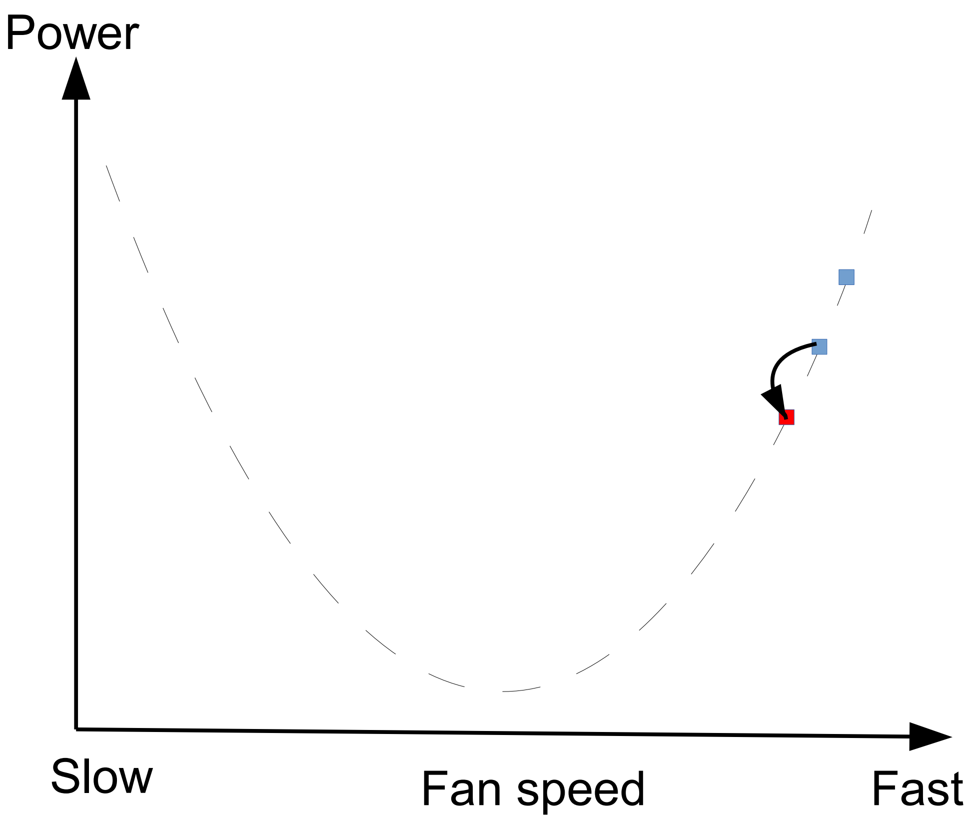
DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting
  - Evaluate close settings



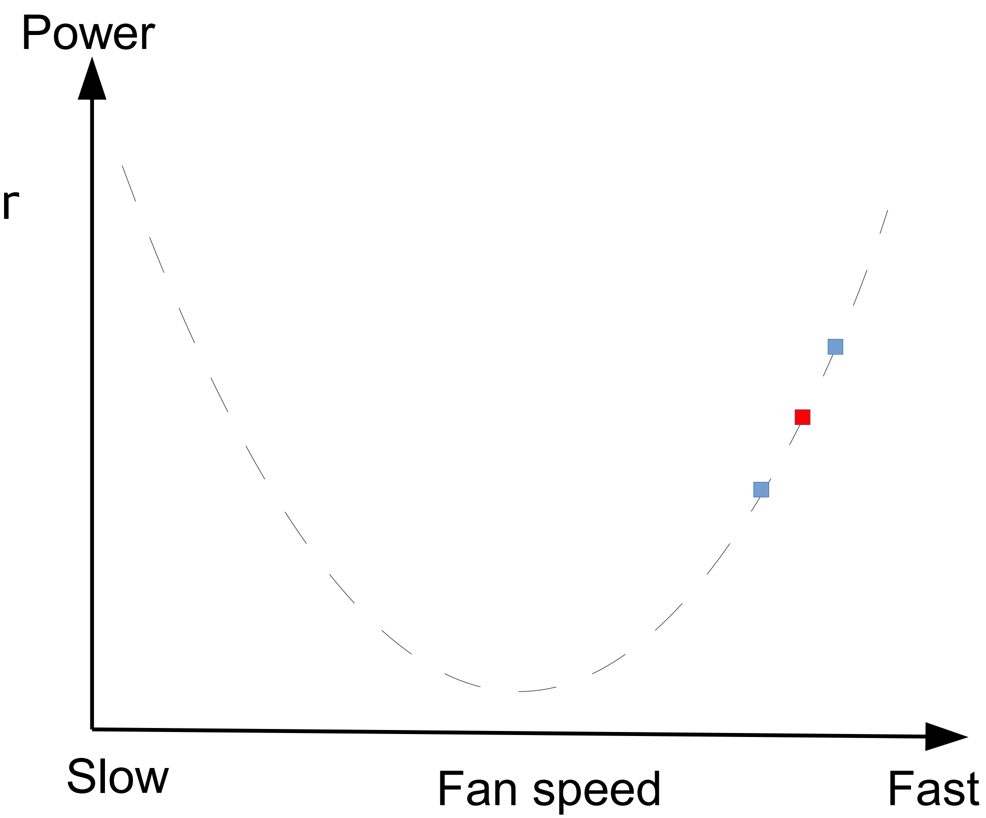
DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting
  - Evaluate close settings
  - Go towards a better setting



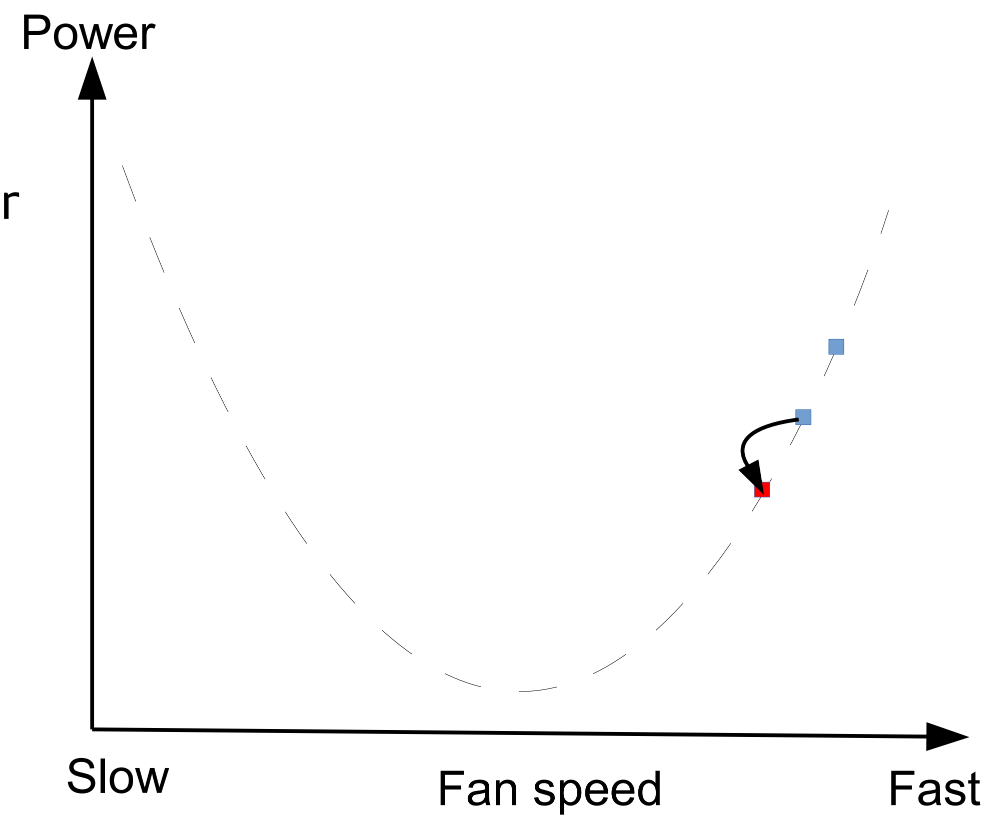
DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting
  - Evaluate close settings
  - Go towards a better setting
  - Start over until no better solution is found



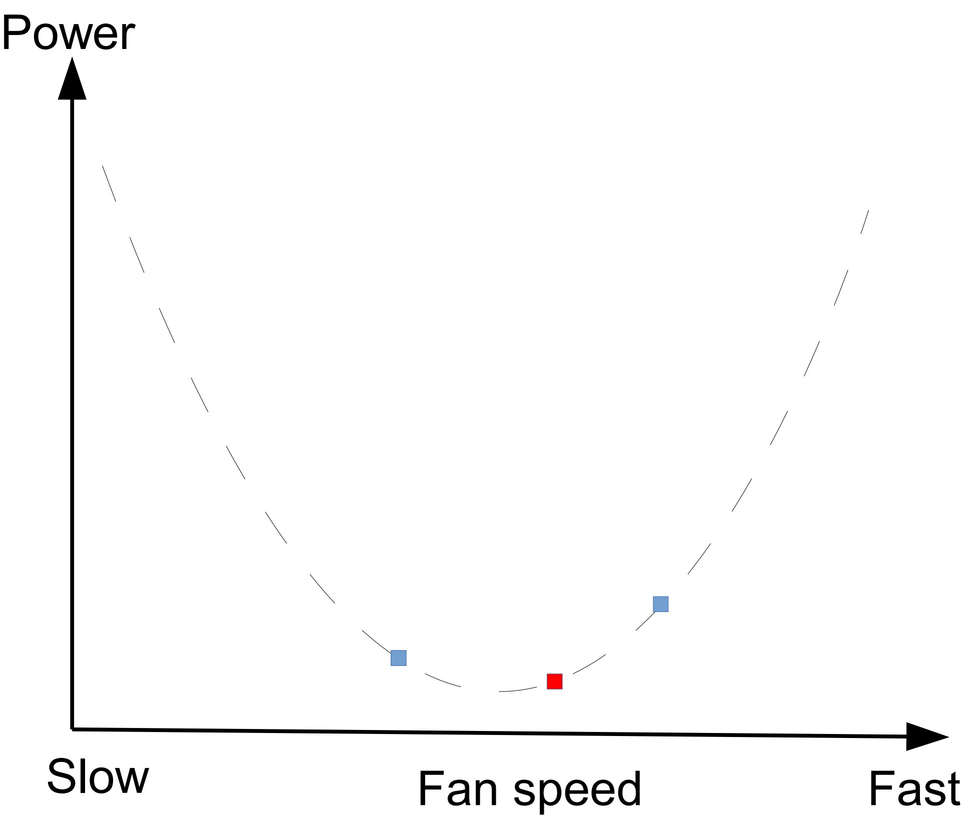
DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting
  - Evaluate close settings
  - Go towards a better setting
  - Start over until no better solution is found



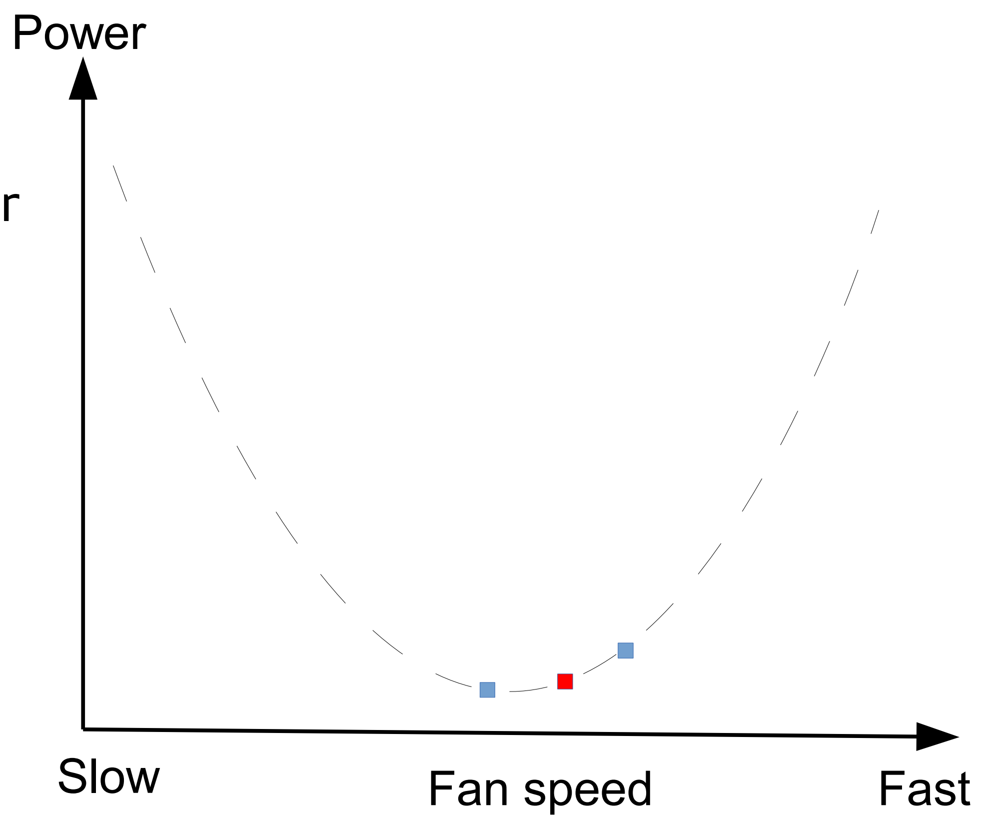
DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting
  - Evaluate close settings
  - Go towards a better setting
  - Start over until no better solution is found



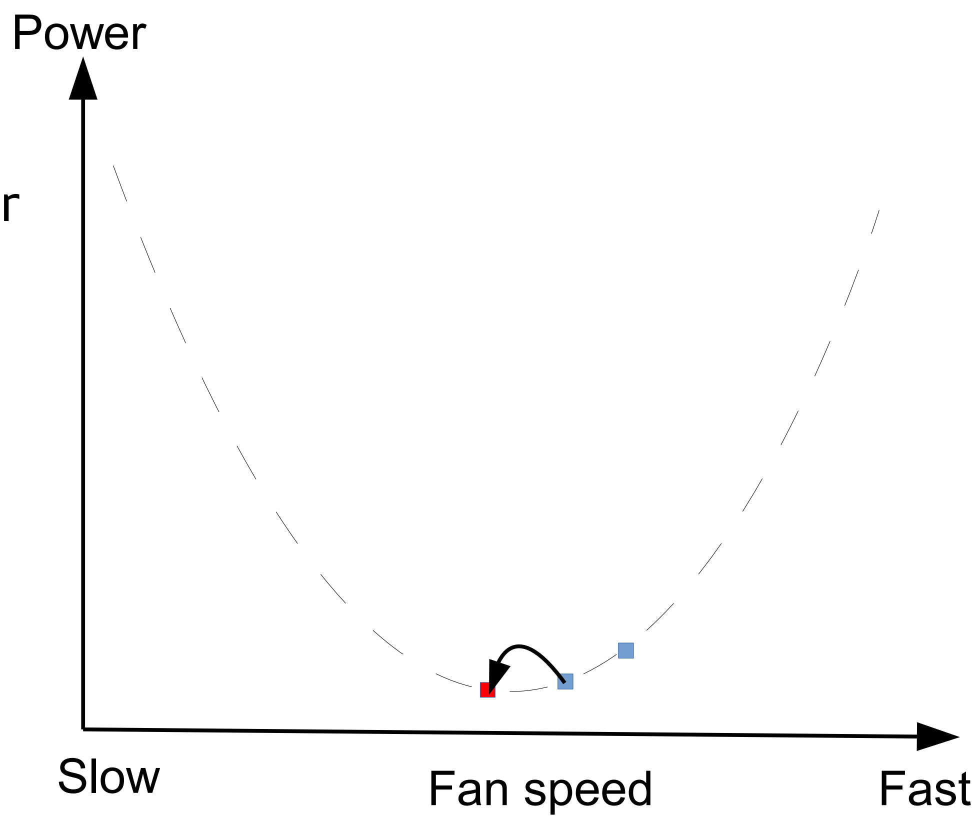
DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting
  - Evaluate close settings
  - Go towards a better setting
    - Or evaluate closer settings
  - Start over until no better solution is found



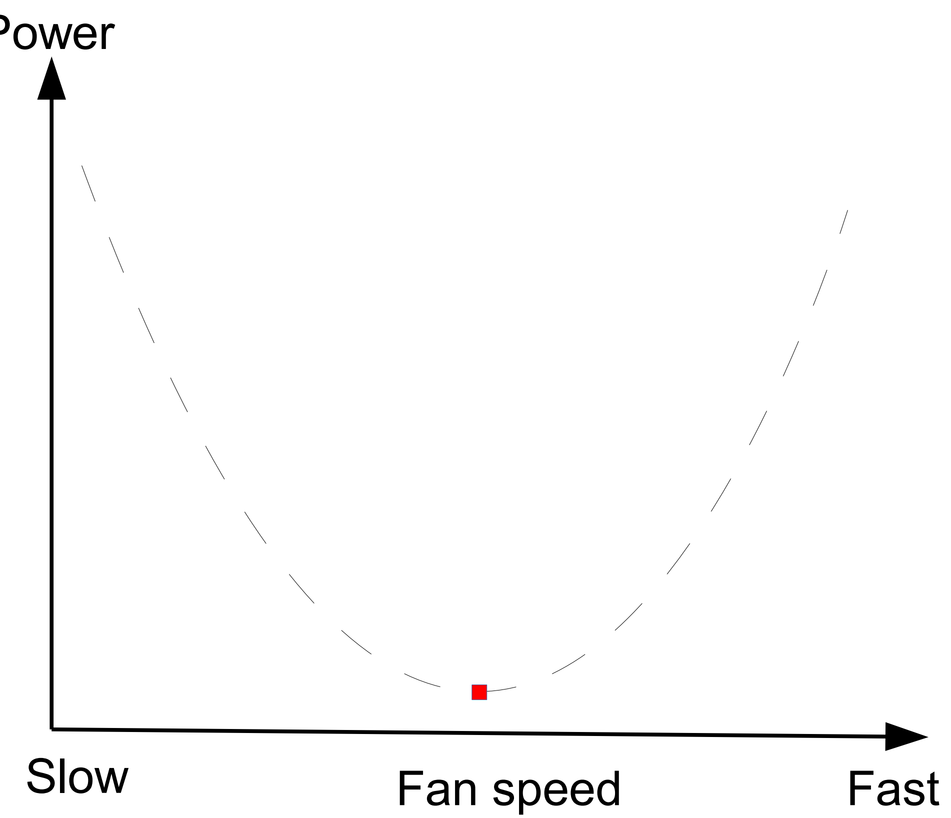
DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting
  - Evaluate close settings
  - Go towards a better setting
    - Or evaluate closer settings
  - Start over until no better solution is found



DFaCE

- Hill climbing:
  - Evaluate power consumption for a given setting
  - Evaluate close settings
  - Go towards a better setting
    - Or evaluate closer settings
  - Start over until no better solution is found



DFaCE

- Hill climbing determines the optimal fan setting
  - Saved for further uses
- Workload impacts heat generation
  - CPU load level approximates heat generation
  - Hill climbing is run for several load levels

DFaCE

- Load level may change during evolution
  - Hill climbing is slow
  - Pause when load level change
  - Resume as soon as possible
    - Last evaluation is lost

- DFaCE is more suited to stable workloads (HPC)

DFaCE

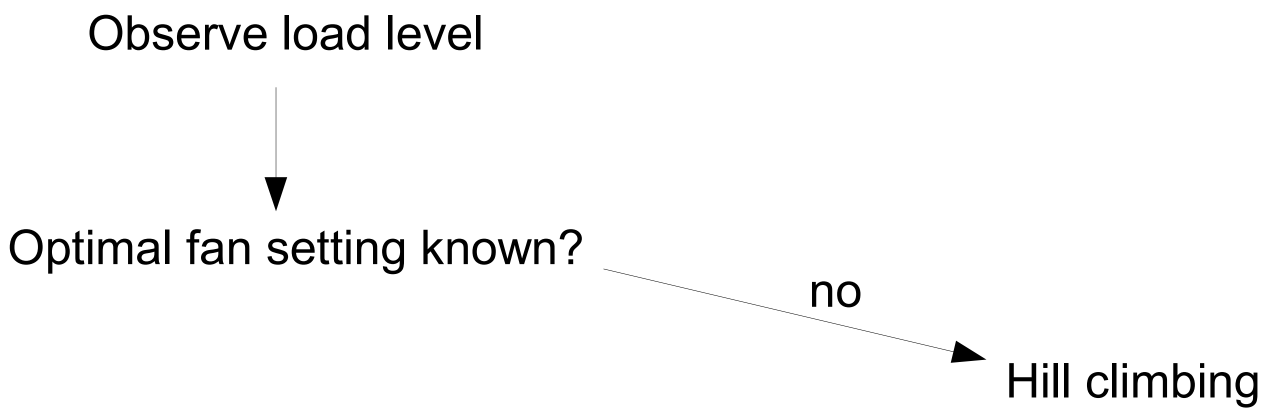
- Temperature may differ from what is expected
  - CPU colder?
    - Stop the fans!
  - CPU much hotter? Hotspots, failures, ...
    - Run again hill climbing

- DFaCE avoids critical temperatures
  - Risks of HW failures

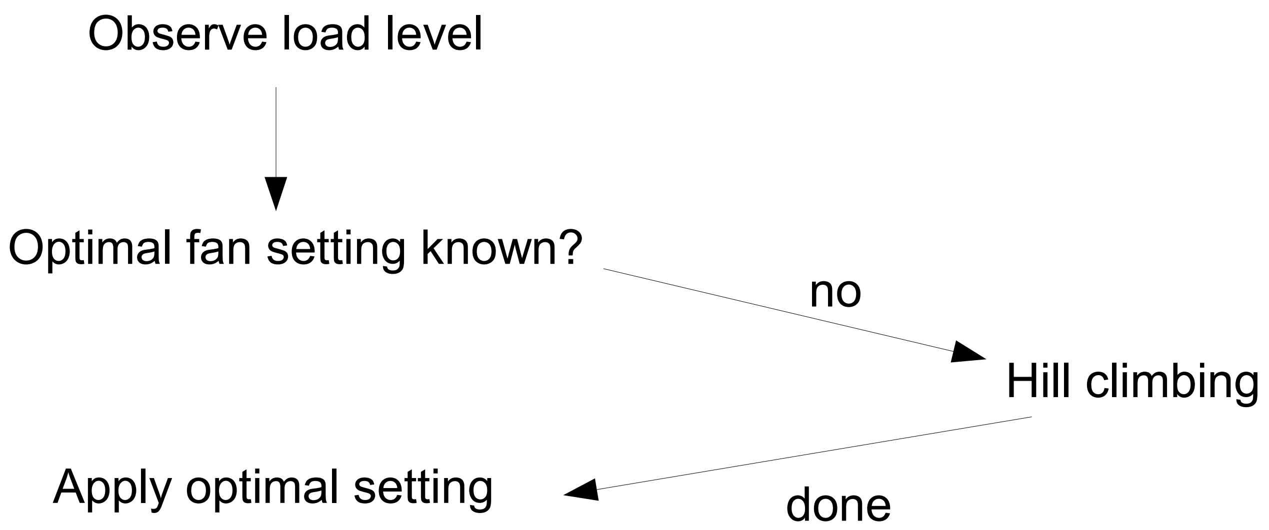
DFaCE

Observe load level

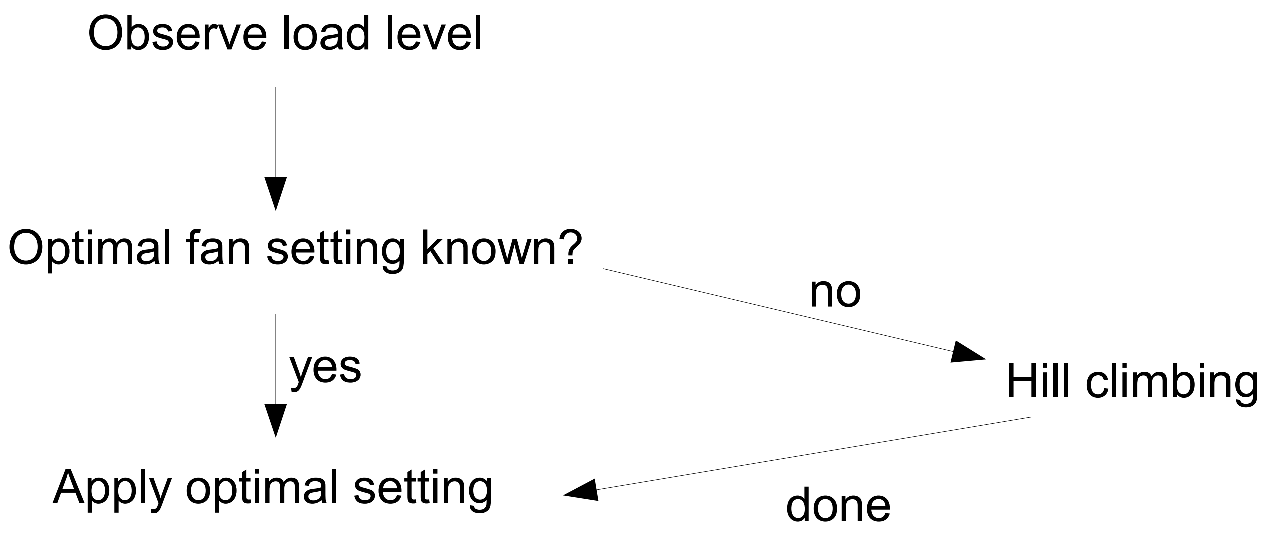
DFaCE


Observe load level
Optimal fan setting known?
no
Hill climbing

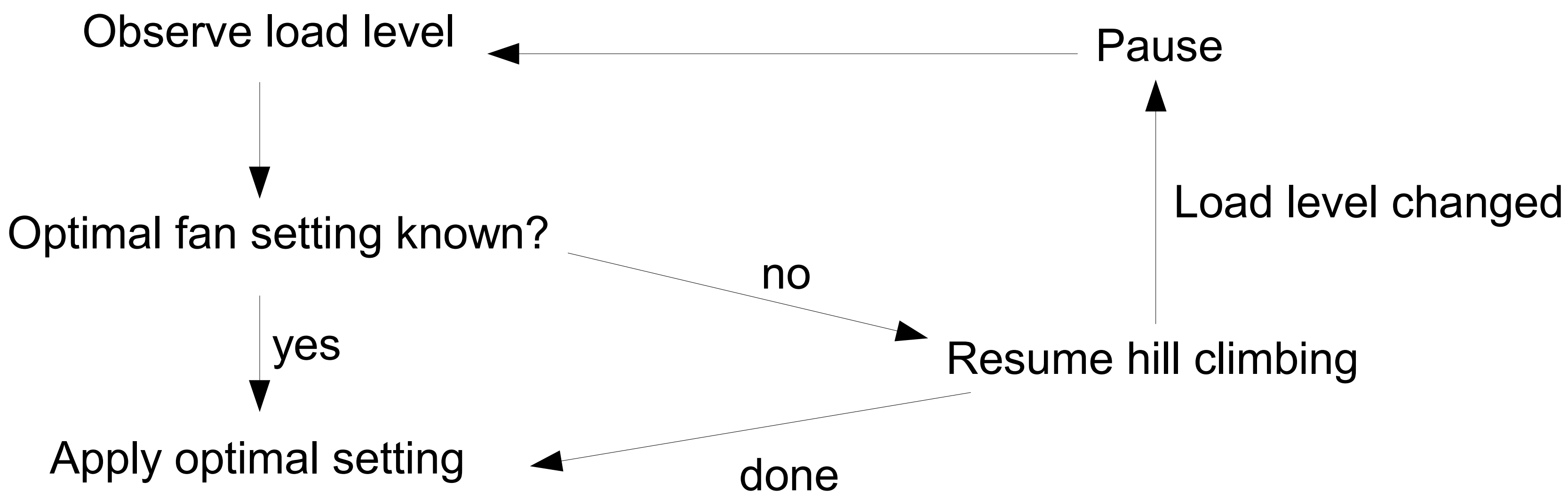
DFaCE


Observe load level
Optimal fan setting known?
no
Hill climbing
Apply optimal setting
done

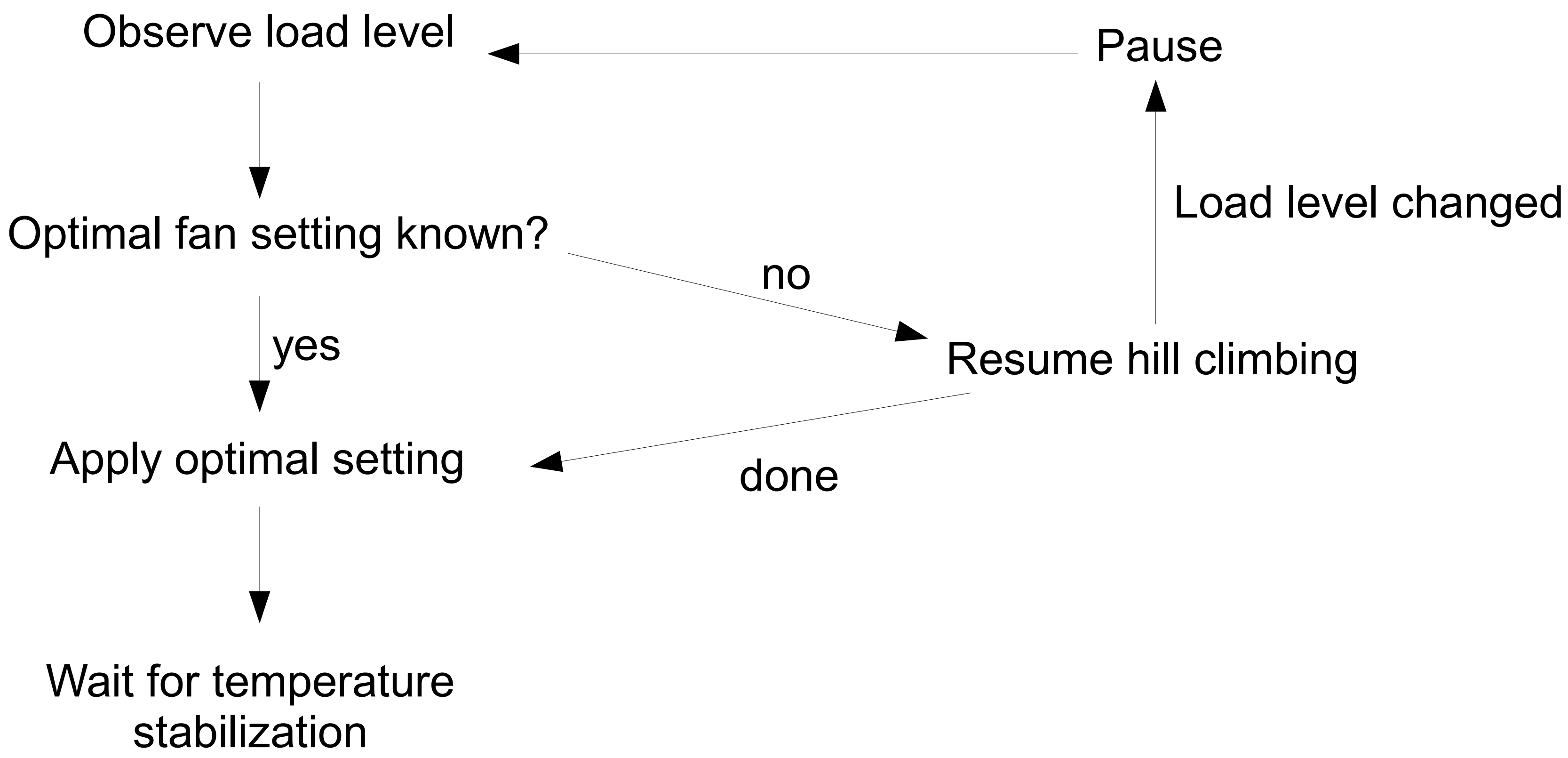
DFaCE


Observe load level
Optimal fan setting known?
no
yes
Hill climbing
Apply optimal setting
done

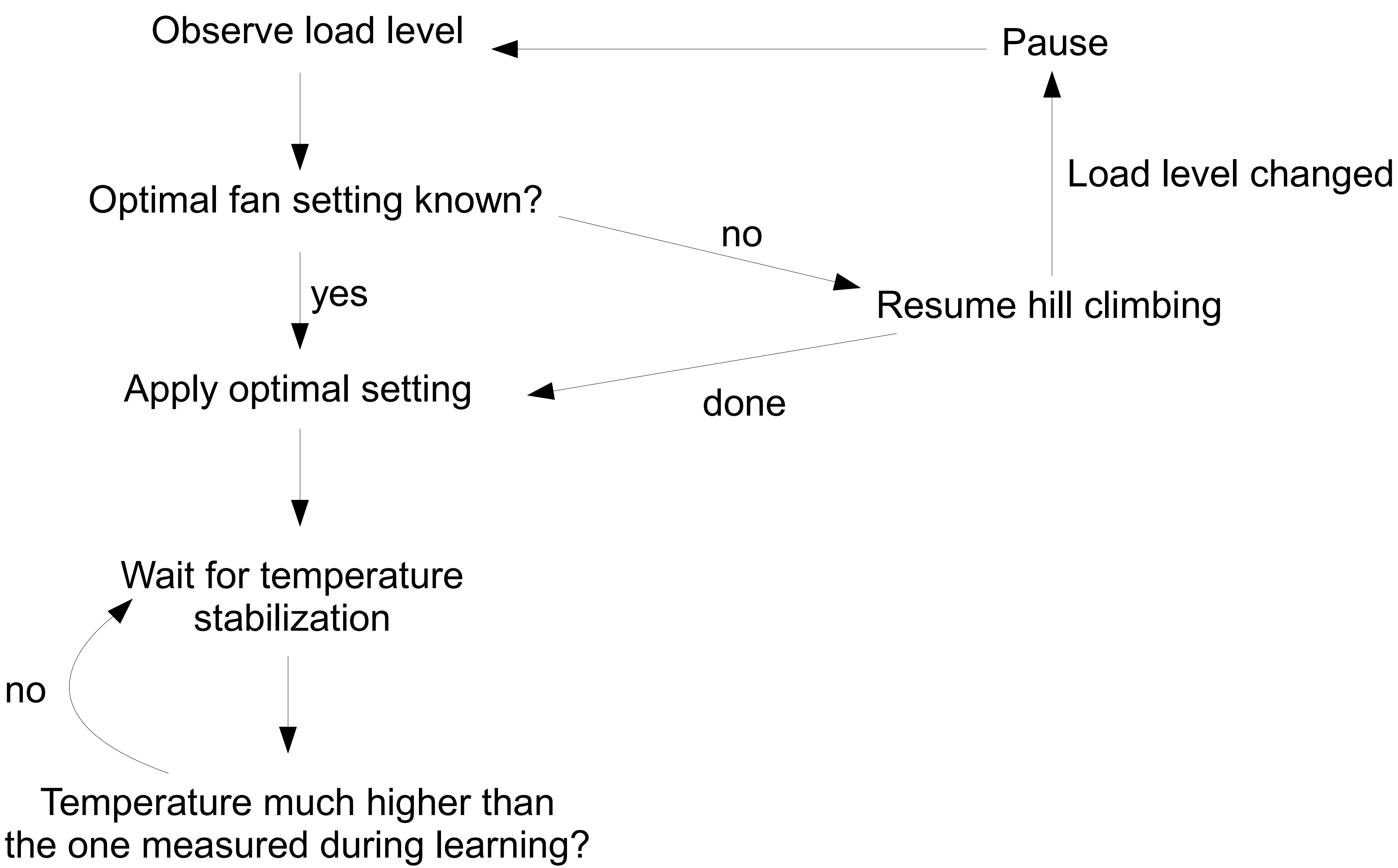
DFaCE


Observe load level
Pause
Load level changed
Optimal fan setting known?
no
yes
Resume hill climbing
Apply optimal setting
done

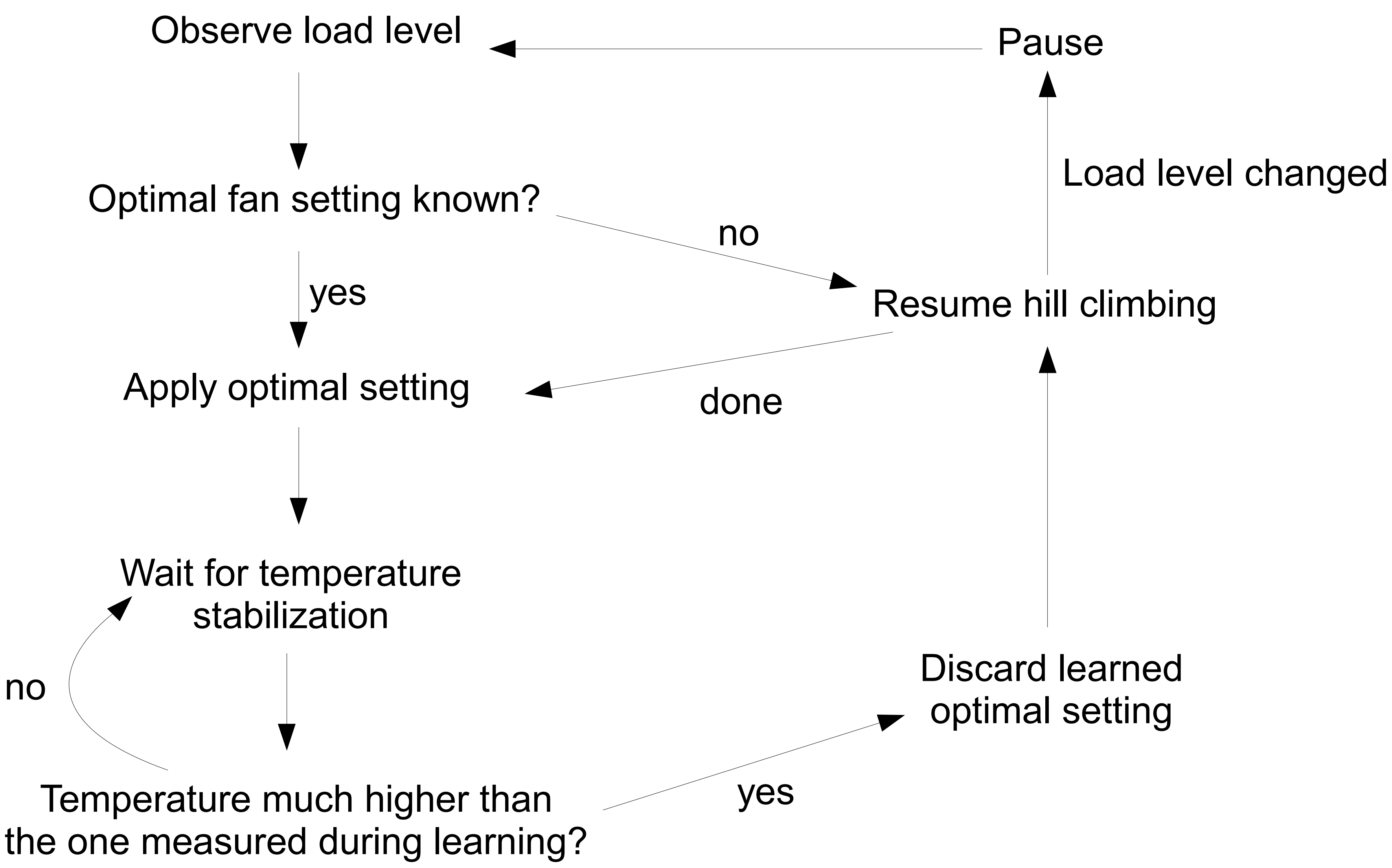
DFaCE


Observe load level
Pause
Load level changed
Optimal fan setting known?
no
yes
Resume hill climbing
Apply optimal setting
done
Wait for temperature stabilization

DFaCE


Observe load level
Pause
Load level changed
Optimal fan setting known?
no
Resume hill climbing
yes
Apply optimal setting
done
Wait for temperature
stabilization
no
Temperature much higher than
the one measured during learning?

DFaCE


Observe load level
Pause
Optimal fan setting known?
no
Load level changed
yes
Resume hill climbing
Apply optimal setting
done
Wait for temperature stabilization
no
Discard learned optimal setting
yes
Temperature much higher than the one measured during learning?

DFaCE

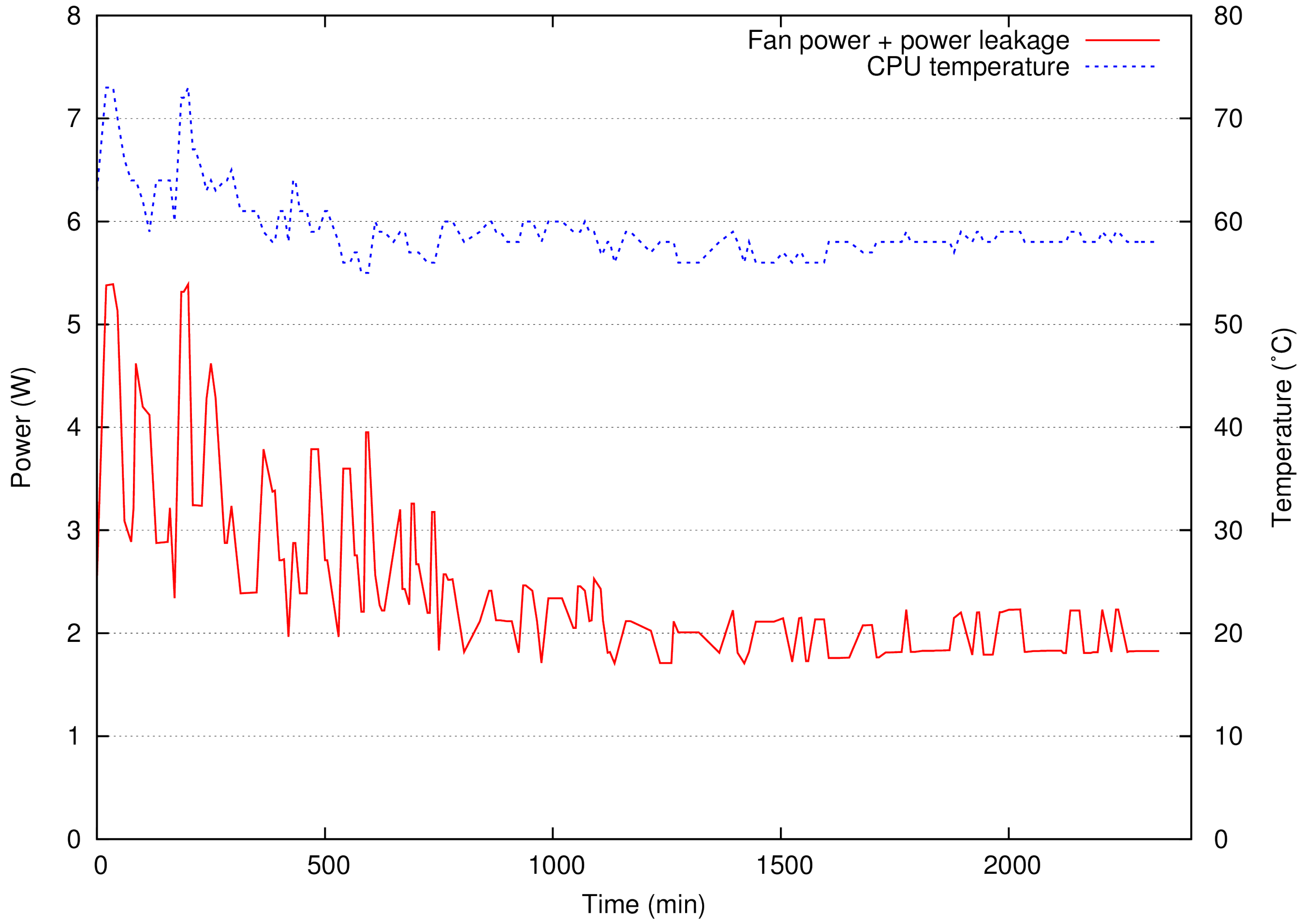
- Hill-climbing applied in background
  - Works as a dynamic system
- Once found, the optimal fan setting is applied
  - Depending on the CPU load level

Experimental results

Experimental results

- Asus P8Z77V PRO motherboard
- Intel Core i5 2380P
- One CPU fan (Scythe Mugen 3)
- Two chassis fans (Alpenföhn Wing Boost 120)

Experimental results



- Slow evolution
- Efficient solutions discovered early

Experimental results

- DFaCE initialized using artificial workloads
  - 0%, 25%, 50%, 75%, 100% learned

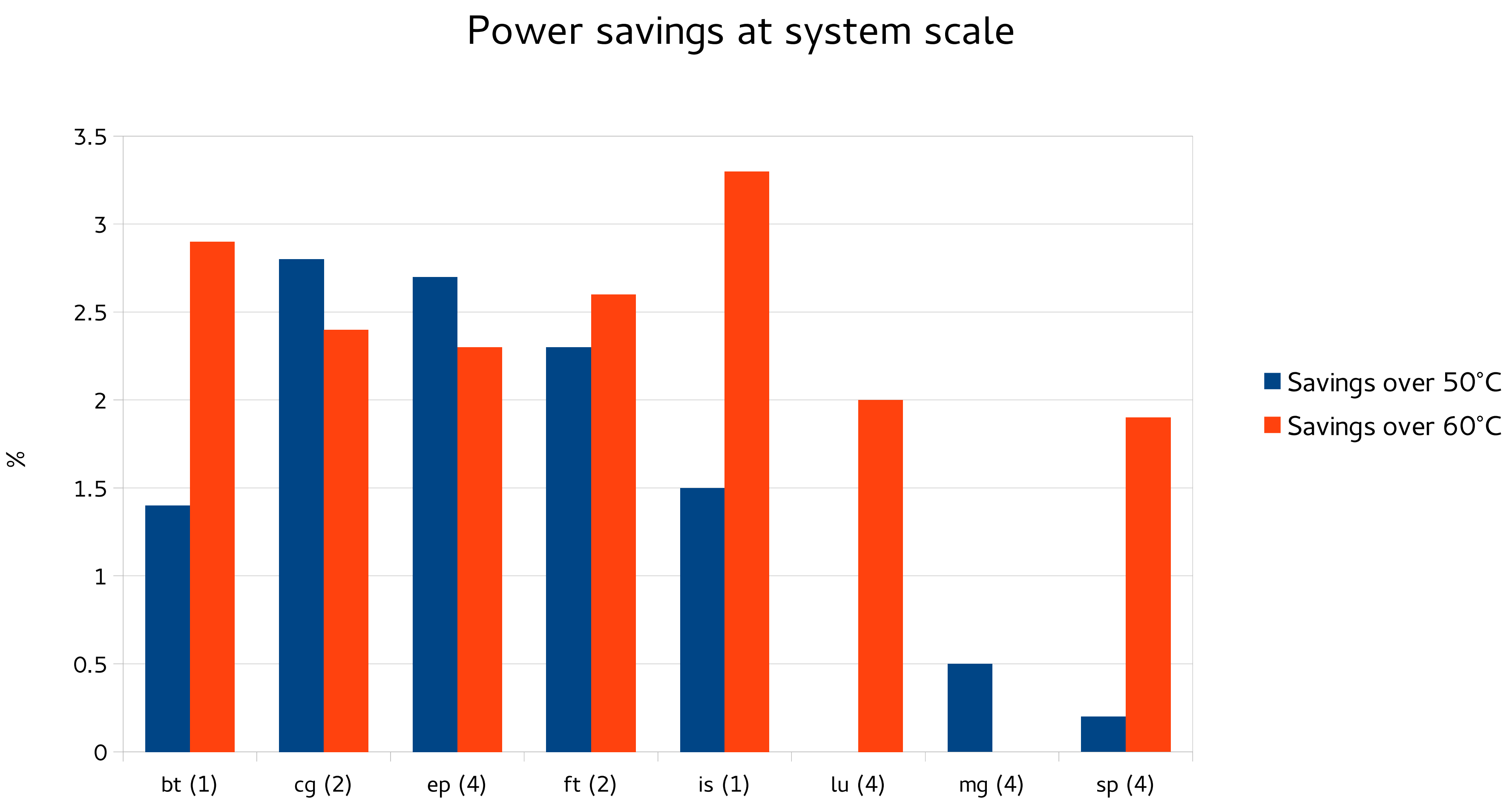
Load level (%)	0	25	50	75	100
Optimal temperature (°C)	36	54	56	64	65

- Target depends on the workload
  - Cannot be achieved by thermal-directed cooling

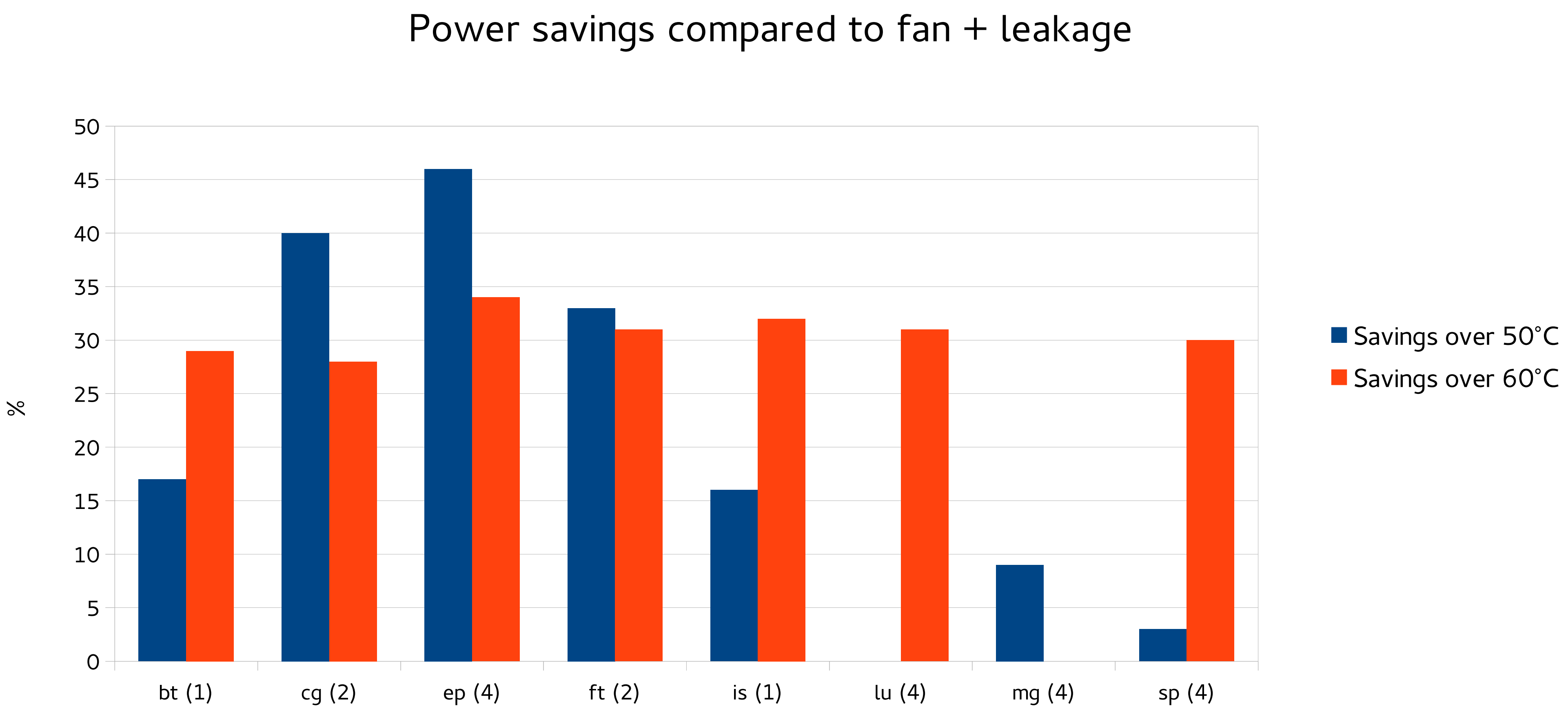
Experimental results

- NAS OMP 3.0 benchmarks (C class)
  - Different number of threads for various load levels
  - 15 repetitions
- Comparison with thermal-directed control
  - 50°C or 60°C targets
- System power measured

Experimental results



Experimental results



Conclusion

Conclusion

- Optimal fan setting learned automatically
  - Energy saved compared to classical controllers
  - Several fans controlled
  - Safety enforced
- Long learning phase (days)
  - Can be accelerated with artificial workloads

Perspectives

- Evaluation on cluster nodes

- Combination with physics models
  - Initializes hill-climbing with realistic setting
  - Accelerates the learning phase
  - Results in the same solution

Thank you for your attention

Any question?