Hewlett Packard Enterprise

Powersched

A HPC System Power and Energy Management Framework

Introduction

Motivation

TOP 500 The List.

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
30	**Hawk** - Apollo 9000, AMD EPYC 7742 64C 2.25GHz, Mellanox HDR Infiniband, HPE HLRS - Höchstleistungsrechenzentrum Stuttgart Germany	698,880	19.33	25.16	3,906

Source: https://www.top500.org/lists/top500/2022/11/

Source: https://www.hlrs.de/solutions/systems/hpe-apollo-hawk

Problem Statement

Overprovisioned Systems + Energy Optimization + No User Input + Heterogeneous Systems

Powersched Framework

Requirements

Overprovisioned Systems

- Holistic view on cluster
- Shift budget between nodes
- Scheduler Integration
- Distribution policies

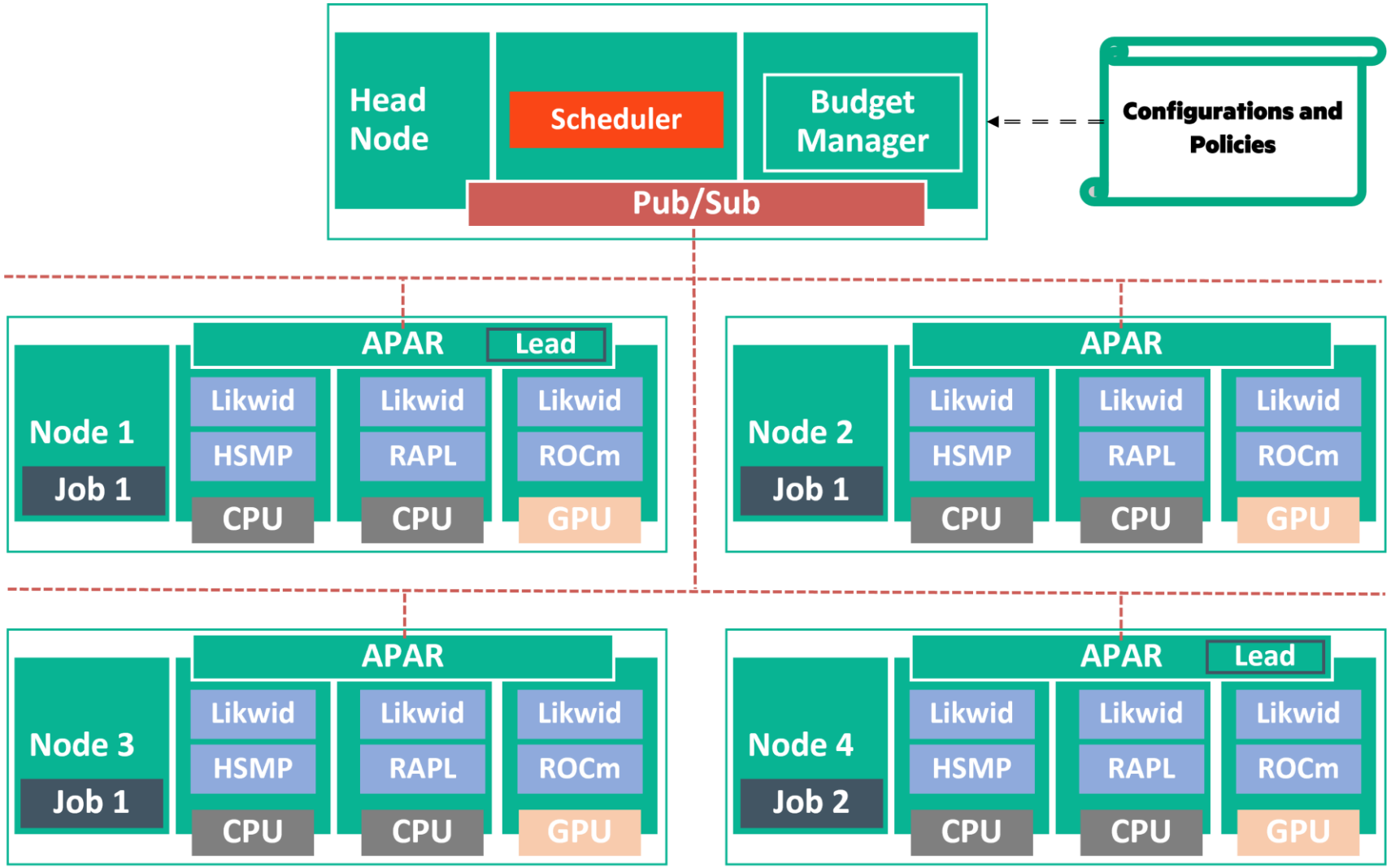
Heterogeneous Systems

- Multiple components in modern systems
- Support diverse landscape

Energy Optimization

- Use system metrics only
- Assign power level with best energy efficiency
- React to different app phases

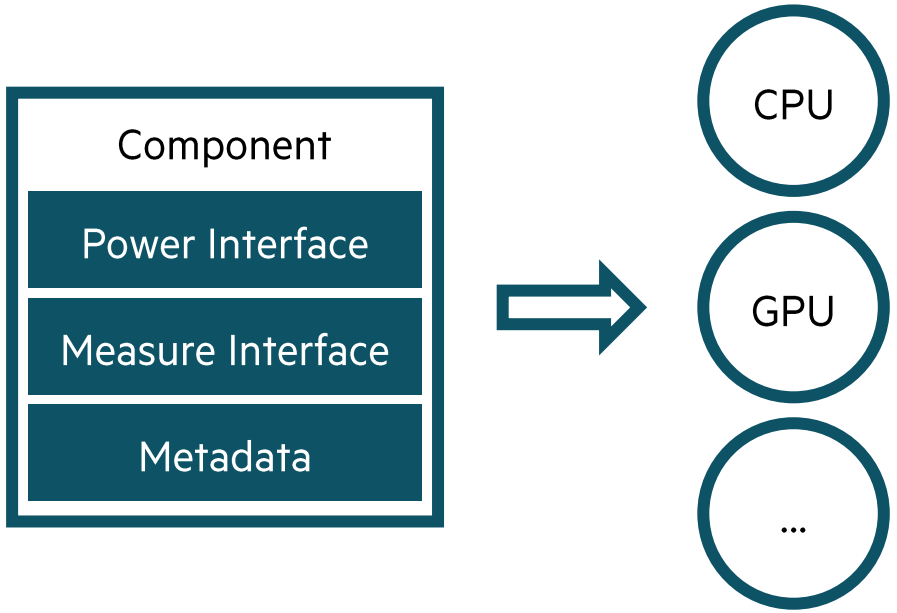
Architecture



Components

Too many components to support individually

→ *can be configured by admin*




```yaml
#----------------------------------------
#  AMD Zen2
#----------------------------------------
parameters:
  socket: int

power_limits:
  min: 100 W
  max: 200 W
  step: 20 W

power_interface:
  type: hsmp
  socket: !param socket

measure_interface:
  type: likwid
  cpu_ids: 0-255
  duration: 120s
  groups:
    - RETIRED_INSTRUCTIONS: PMC1
      RETIRED_SSE_AVX_FLOPS_ALL: PMC2
      RETIRED_BRANCH_INSTR: PMC3
      RETIRED_MISP_BRANCH_INSTR: PMC4
      RAPL_PKG_ENERGY: PWR1

    ...
```

Failure Tolerance and Documentation

Redundancy

- Independent out-of-band control mechanism via mpower.
  - Aggregates PDU power per power domain (UPS)
  - Runs on hawk-master, aggregated PDU data from TimescaleDB (Hawk-Monitor4)
  - Overrides PowerSched settings in case of detected inconsistencies.

Failure tolerant communication and control

- System transitions to the following state if either a power head-node crash or a communication failure occurs.
  - Running jobs remain in status quo. If the head-node database can be recovered running jobs will be reset to an optimal value. If the database cannot be recovered the jobs will keep running on idle power.
  - Exiting jobs transition to idle power.
  - New starting jobs will remain on idle power until the head-node recovers.

Installation and further Documentation

https://kb.hlrs.de/admin/index.php/HPE_System_administration#Power_Management_(PowerSched)

Energy Optimization – Golden Section Search

Direct Optimization of Governing Metric

Steady-state assumption – constant averaged instruction stream, typical +100s of workload iterations.

- Long measurement intervals of order minutes ⟶ zero-ish overhead in application runtime.

We measure:

- Energy per package
- Retired instructions/sec - mapped to application performance

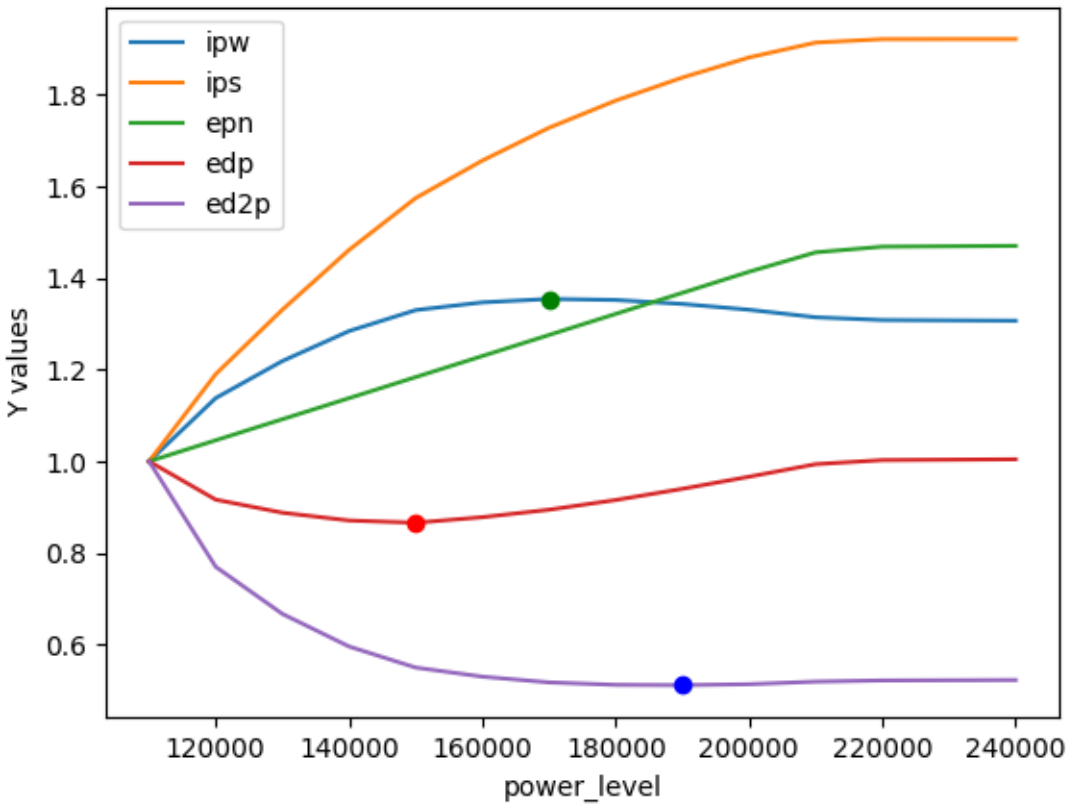
  1/[Retired instructions/sec] – mapped to **normalized application runtime** („delay“)

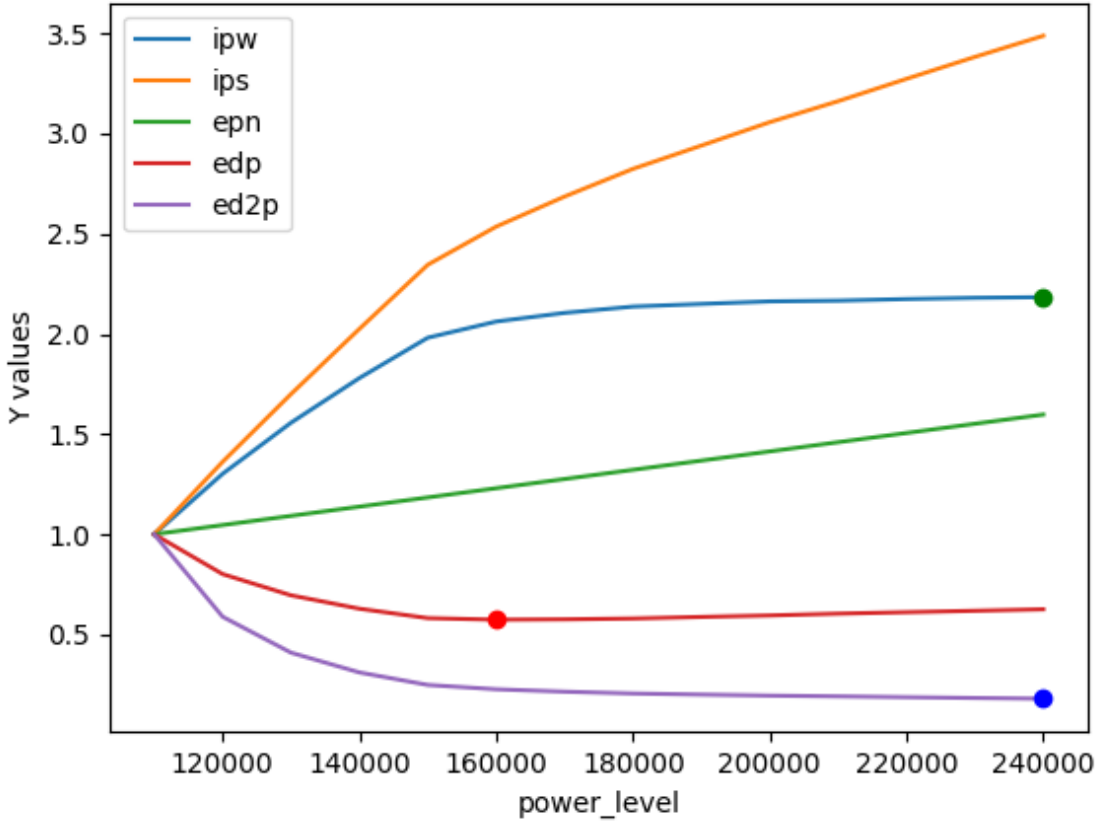
⟶ we hence can directly optimize a governing metric such as an **energy-delay product** without having to rely on fragile mechanisms such as LD_PRELOAD.

- Can run transparently, policy guided, no user intervention required, requires no ML training, works for corner cases, expandable to support heterogeneous architectures, GPU.

Direct Optimization of Governing Metric

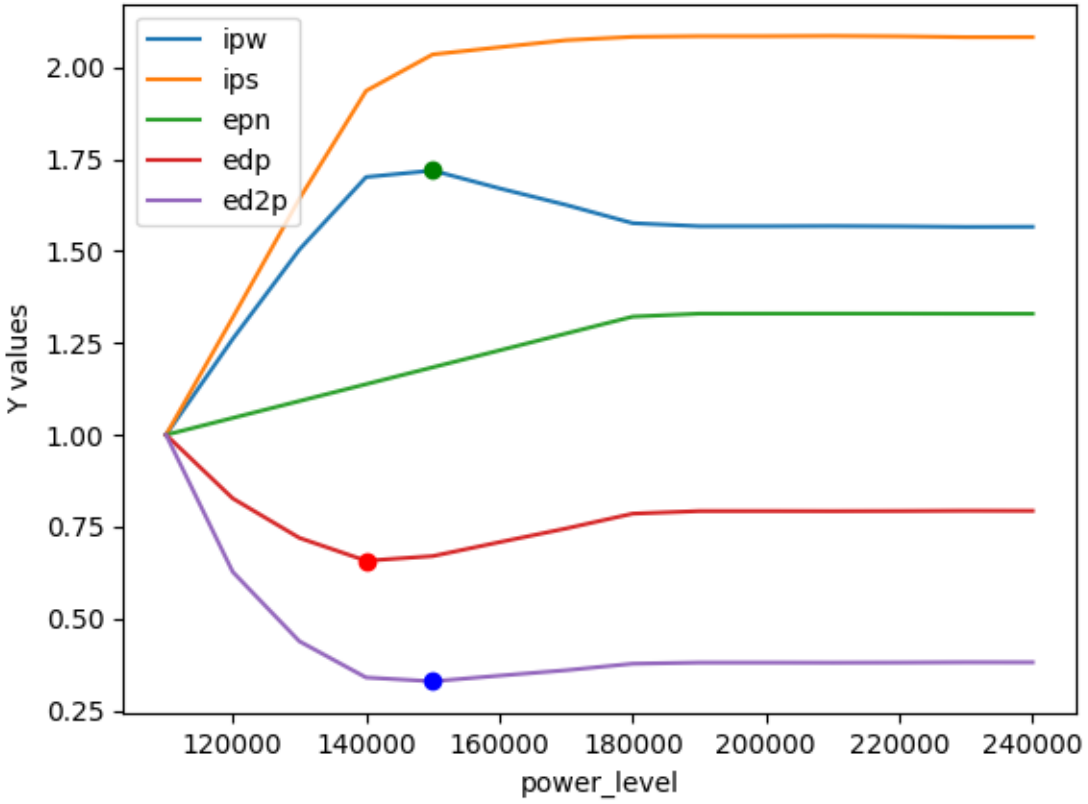
A comparison of EDP, ED2P

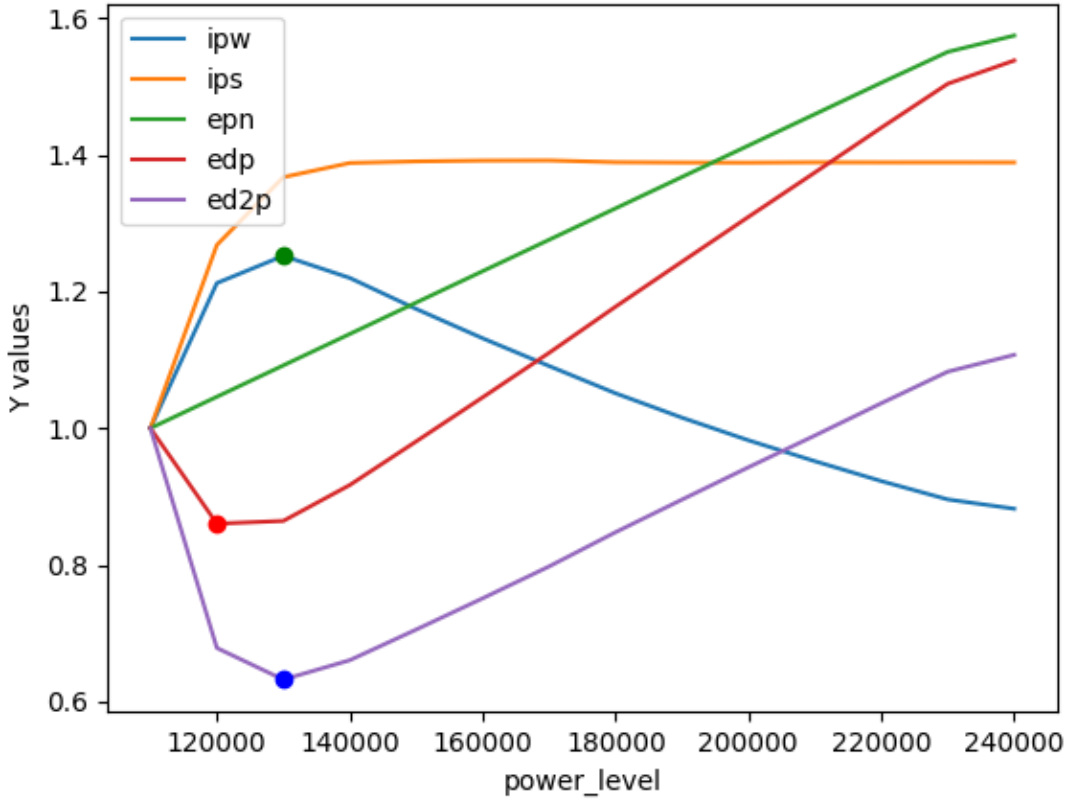




Direct Optimization of Governing Metric

A comparison of EDP, ED2P





Direct Optimization of Governing Metric

Observation 1:

- Minima can be narrow. Selecting pre-defined frequencies won‘t do.

Observation 2:

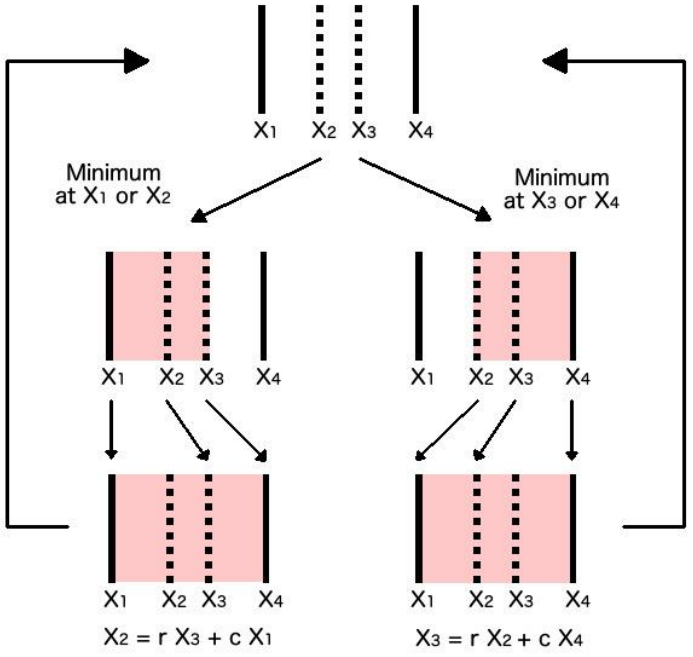
- Energy-delay product metric is too complex to fit a nonlinear expression with a sparse and noisy data set.

- We determine the minimum of the ED2P metric via an iterative algorithm, which converges fast, but does not require the calculation of derivatives.

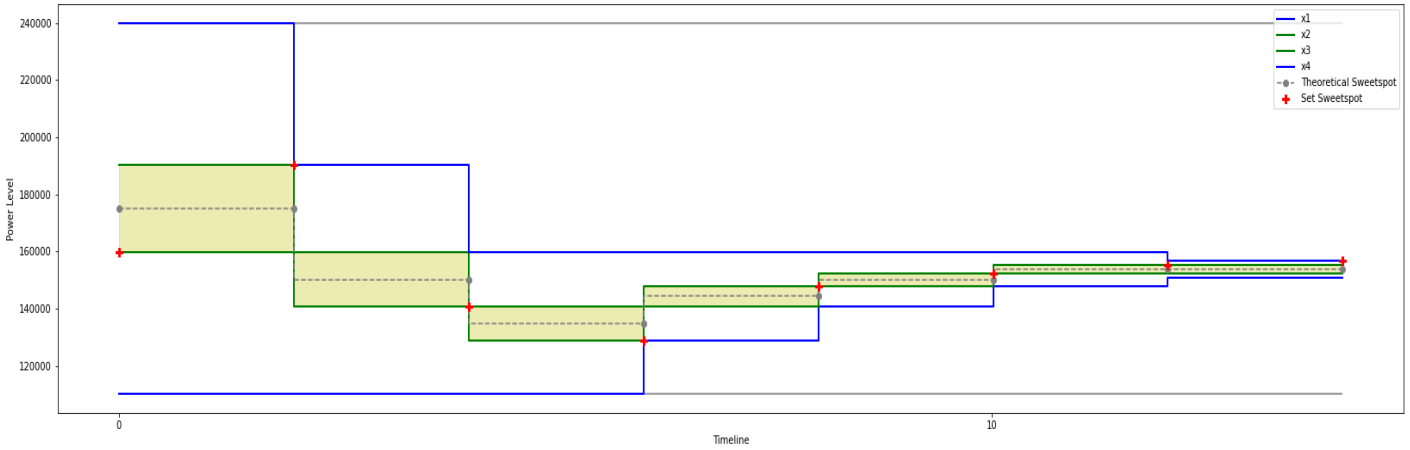
Golden Section Search

- Iterative narrowing of search space

Power limits are related by golden ratio
→ allows reusing values from previous iterations



https://upload.wikimedia.org/wikipedia/commons/6/68/Diagram_of_a_golden_section_search.jpg

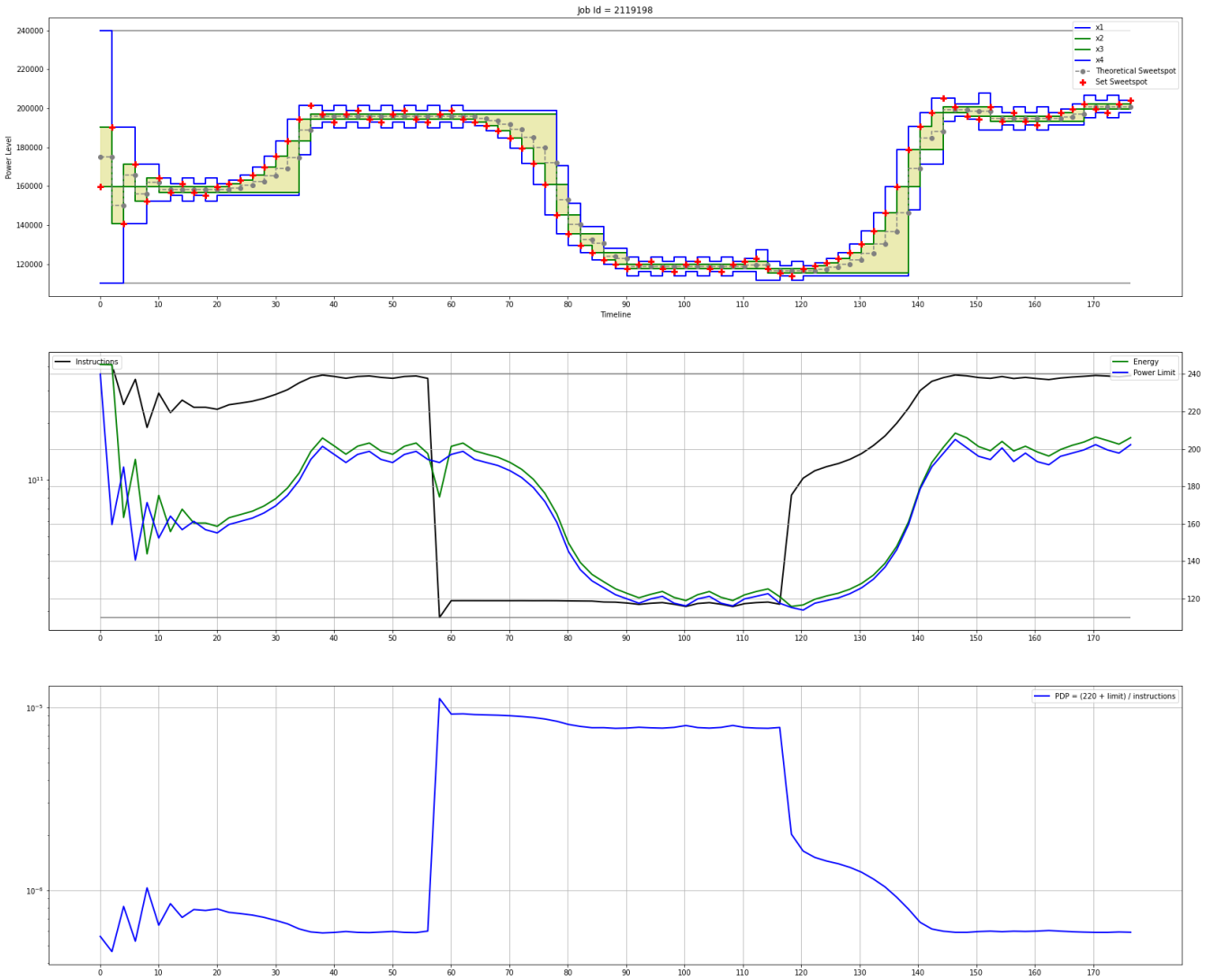


Reacting to different application phases

- Implement “inverted” golden section search

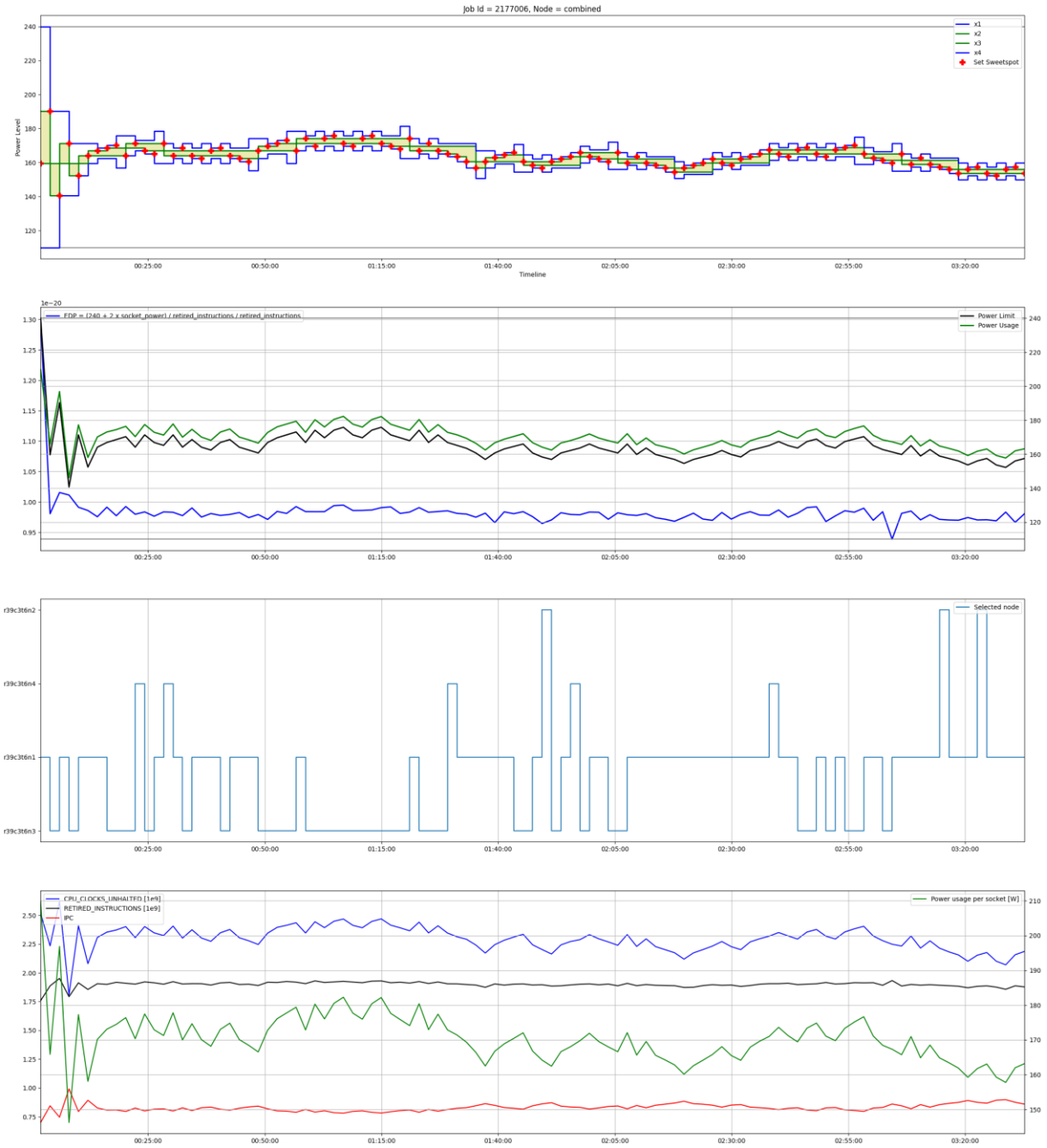
→ exponential step size increase

→ approach limits, without exceeding them

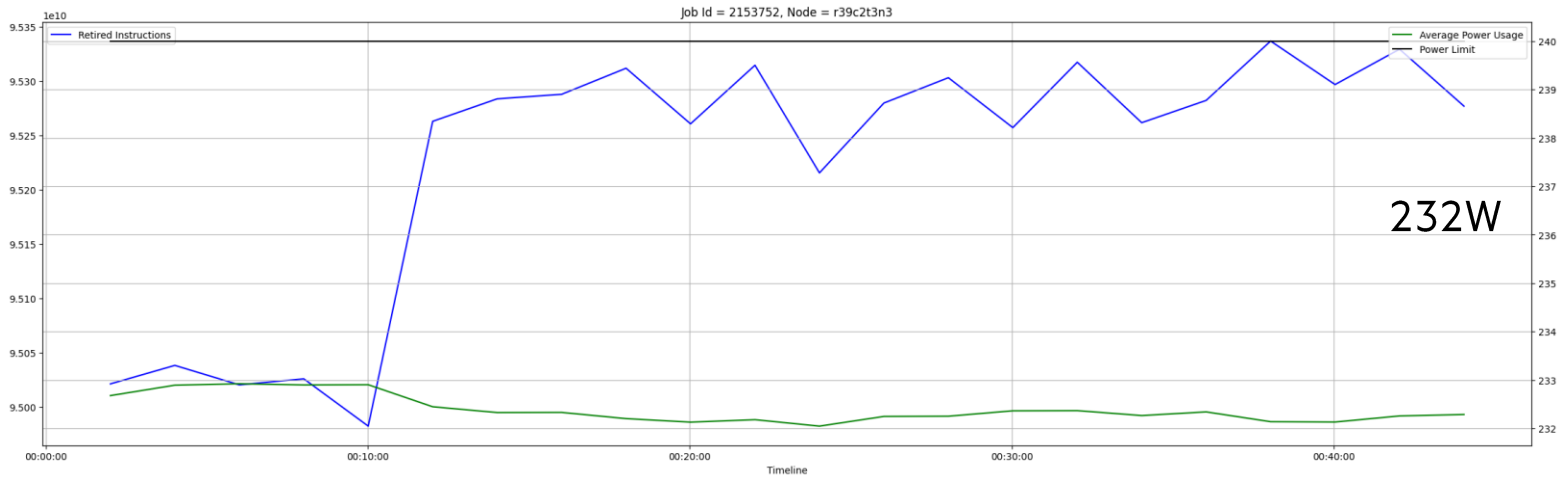


Evaluation

- GSS leader determined by max. retired instructions.

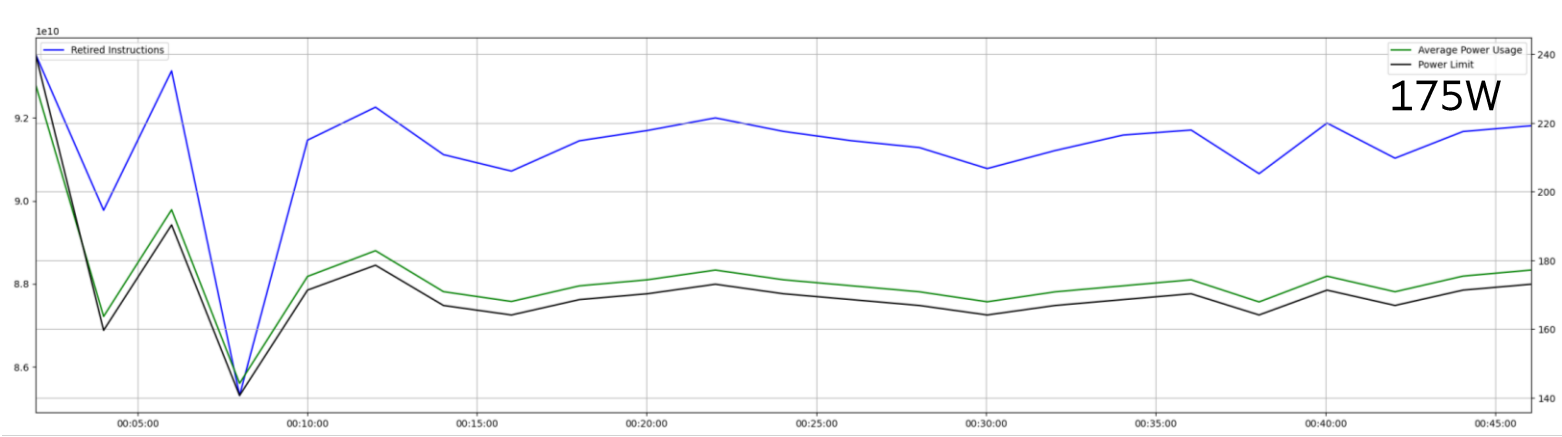


Evaluation



NS3DNeo - onegas 16 Nodes

- 240W per socket



NS3DNeo - onegas 16 Nodes

- PowerSched, dynamic steering

Performance degradation

- +3.4%

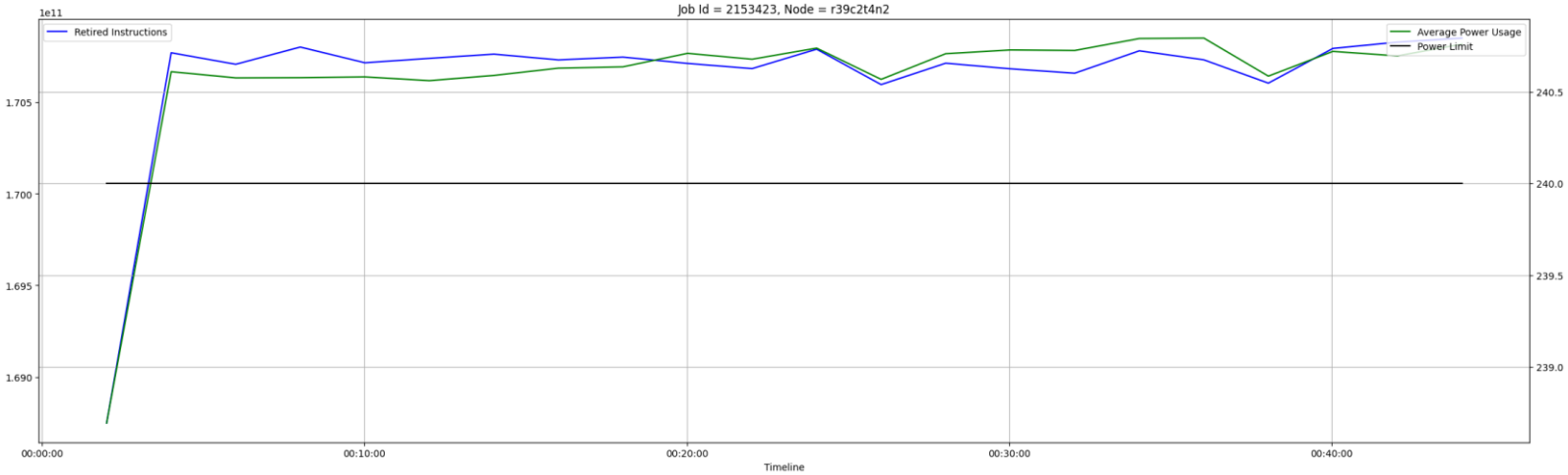
Energy consumption

- -16,2%

(F + 2 x ENERGY_PER_SOCKET)
/ RET_INST_PER_SEC^2

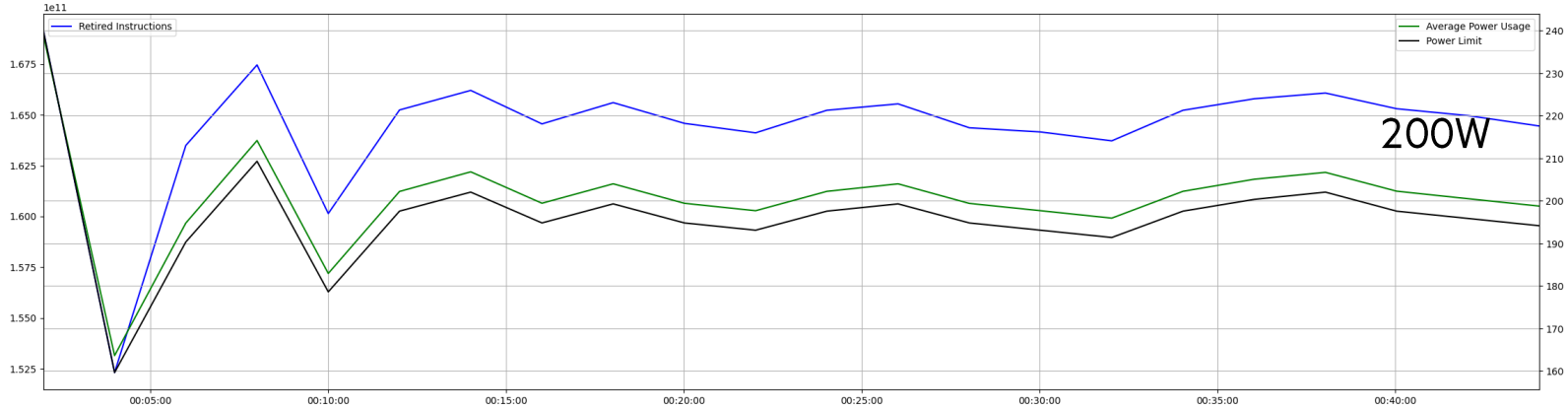
Evaluation

240W



NS3DNeo - twogas 8 Nodes

- 240W per socket



NS3DNeo - twogas 8 Nodes

- PowerSched, dynamic steering

Performance degradation

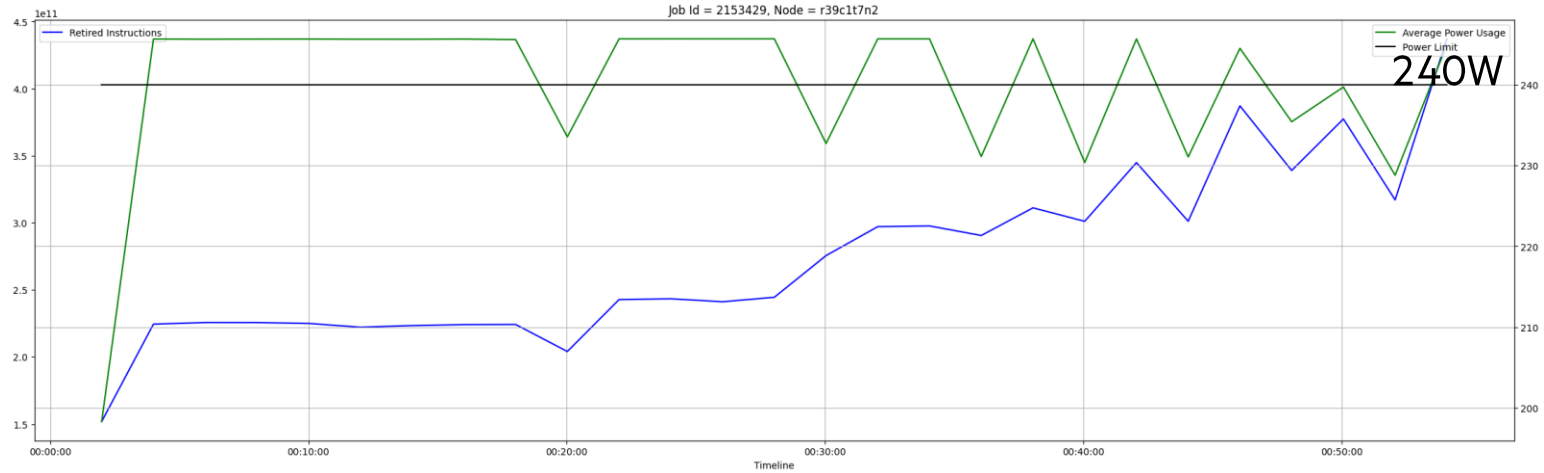
- +3.3%

Energy consumption

- -11,4%

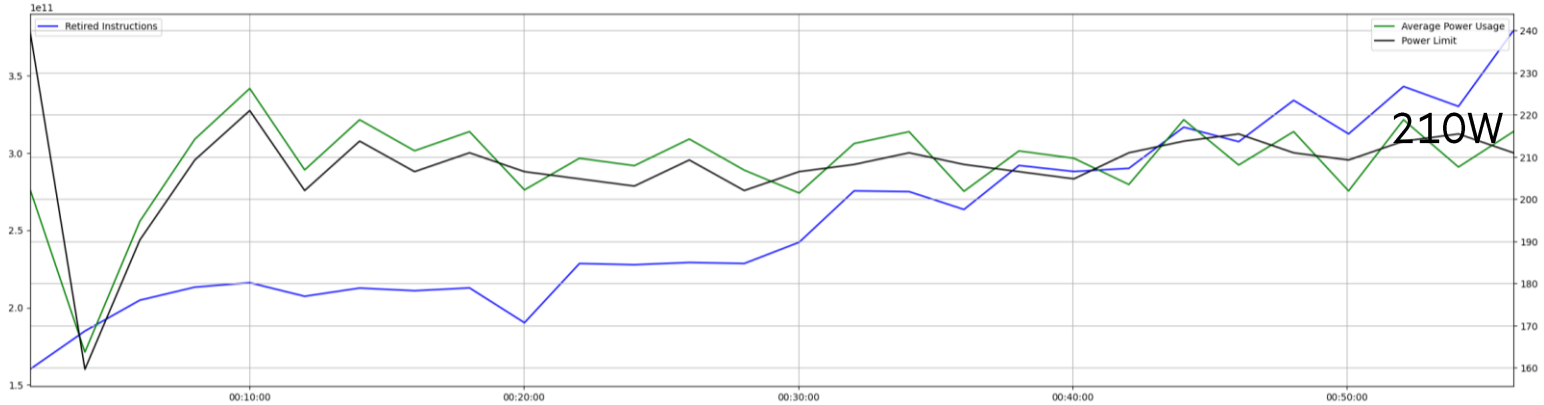
(F + 2 x ENERGY_PER_SOCKET)
/ RET_INST_PER_SEC^2

Evaluation



StarCCM+ 16 Nodes

- 240W per socket



StarCCM+ - 16 Nodes

- PowerSched, dynamic steering

Performance degradation

- +1%

Energy consumption

- -8,5%

(F + 2 x ENERGY_PER_SOCKET)
/ RET_INST_PER_SEC^2

Evaluation

Application	Use Case	PD [%]	PE: MAX [W]	PE: PS [W]	EPS [%]	EPN [%]
Flexi	Freestream, Balanced, 16 Nodes	1,70%	240,00	195,00	-18,75%	-14,75%
Flexi	Freestream, Imbalanced, 16 Nodes	1,20%	230,00	225,00	-2,17%	-1,49%
LBC	Lid-driven Cavity, 32 Nodes	3,80%	225,00	175,00	-22,22%	-17,54%
NS3DNeo	Multi-Phase Flow, TG, 8 Nodes	4,10%	238,00	195,00	-18,07%	-14,10%
NS3DNeo	Multi-Phase Flow, TG, 16 Nodes	2,70%	227,00	200,00	-11,89%	-8,71%
NS3DNeo	Multi-Phase Flow, TG, 32 Nodes	0,60%	197,00	195,00	-1,02%	-0,66%
NS3DNeo	Multi-Phase Flow, ZPG, 8 Nodes	3,30%	240,00	200,00	-16,67%	-12,90%
NS3DNeo	Single-Phase Flow, 16 Nodes	3,40%	238,00	175,00	-26,47%	-22,11%
NS3DNeo	Single-Phase Flow, 8 Nodes	2,20%	235,00	170,00	-27,66%	-23,21%
OpenFoam	FixedTol, Single-Phase OpenMPI, 8Nodes	-2,70%	240,00	225,00	-6,25%	-4,48%
OpenFoam	FixedTol, Single-Phase, MPT, 8Nodes	-3,50%	240,00	200,00	-16,67%	-12,90%
OpenFoam	Weir, Two-Phase, MPT, 8Nodes	1,00%	235,00	170,00	-27,66%	-23,21%
OpenFoam	Weir, Two-Phase, OpenMPI, 8Nodes	-0,30%	240,00	220,00	-8,33%	-6,06%
StarCCM+	Open Water Propeller, 16 Nodes	1,00%	240,00	205,00	-14,58%	-11,11%
XHPL	HPL-2.3, 8 Nodes	2,00%	240,00	235,00	-2,08%	-1,45%

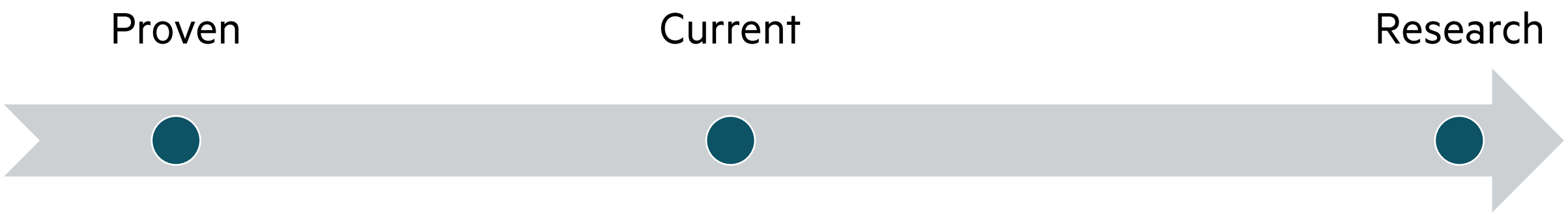
Summary & Outlook

Summary & Outlook

Extensible framework that supports multiple user-defined components

Clustering approach that can **save up to 14% energy** with only 2% runtime increase

Summary & Outlook



Elimination of training time with
Golden Section Search

GSS approach that can **save than 23% energy** with only 2% runtime increase, never exceeds 4% performance degradation.

Summary & Outlook

User-Report

Policy Support

- Other optimization approaches, reinforced learning
- Multi-dimensional optimization for heterogeneous architectures, GPU
- Investigating MPI issues, goal: saving more energy in (strongly) imbalanced applications, strong scaling, OpenMPI

Thank you!

Marcel Marquardt
marcel.marquardt@hpe.com

Jan Mäder
jan.maeder@hpe.com

Christian Simmendinger
christian.simmendinger@hpe.com

Torsten Wilde
wilde@hpe.com

Tobias Schiffmann
tobias.schiffmann@hpe.com

© 2023 Hewlett Packard Enterprise Development LP