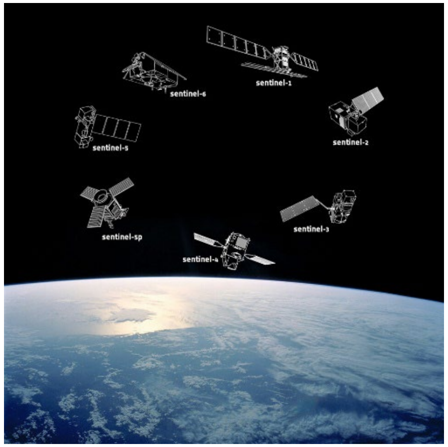
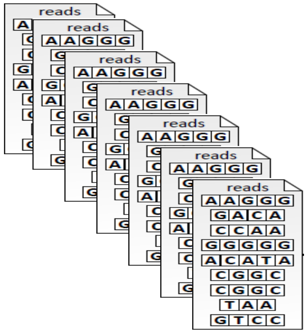
Seminar Saving Energy in (Large-Scale) Data Analysis

Ulf Leser, Fabian Lehmann

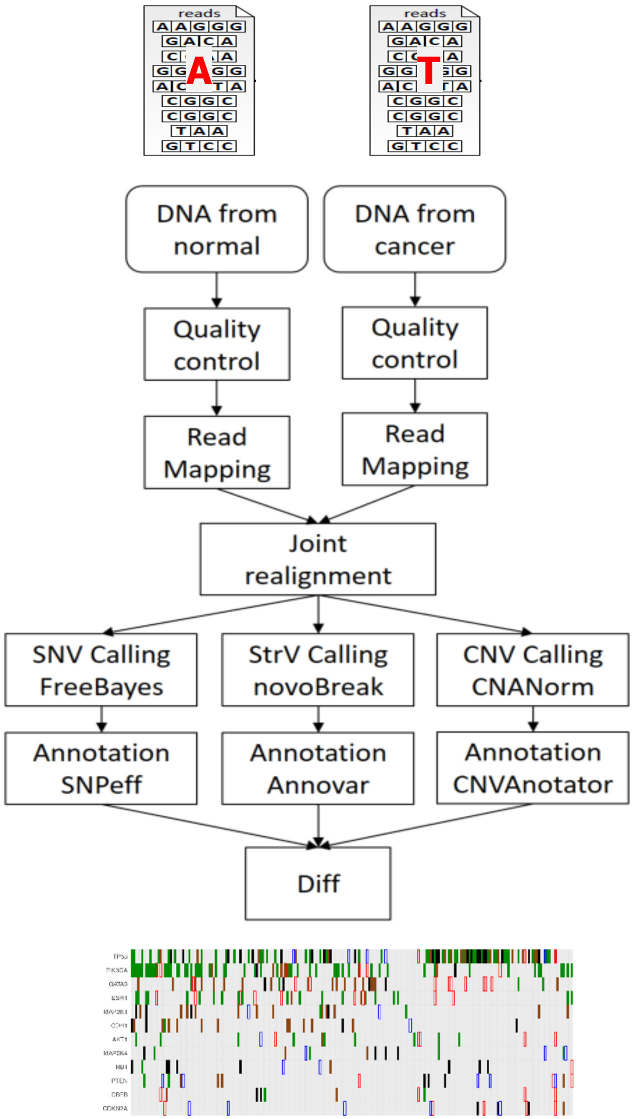
Big Scientific Data

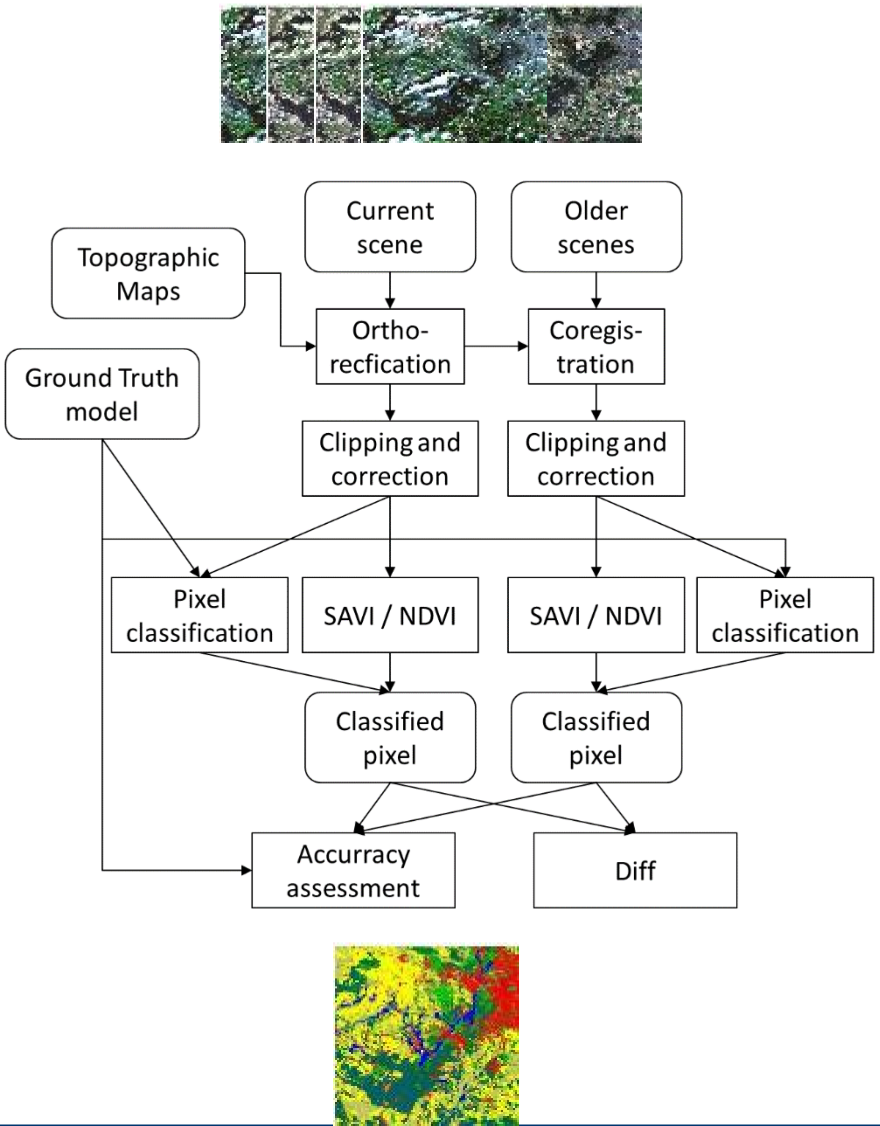




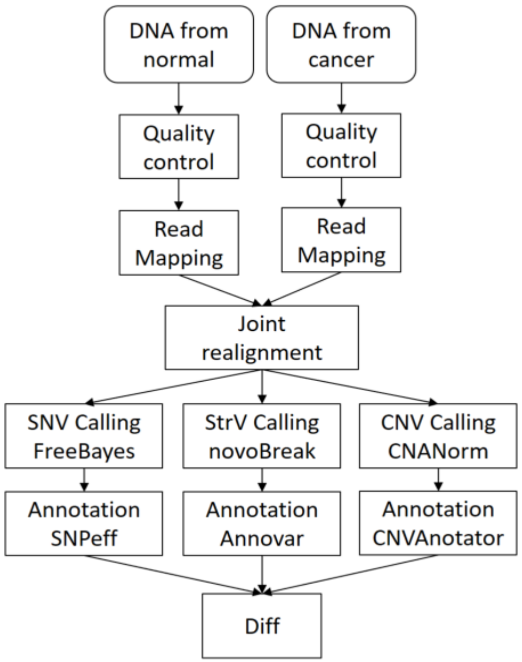
Not petabytes every day, but easily a few terabytes per week

Data Analysis Workflows (DAWs)





Distributed DAW Infrastructure



Reducing Runtime: High-Performance Computing

But

Examples

- Next generation of Nvidia GPUs is expected to consume more power per year than the Netherlands [Heise, 2023]
- Compute centers around the world consumed 500 - 650 twh in 2021 – about as much as all of Germany [Deutschlandfunk, 2022]
- German compute centers consume 10 twh in 2010, 18 twh in 2022, and could consume 35t wh in 2030 [Borderstep-Institut]
  - ~1% of German electricity requirements

HU Scale: HPC@HU

- 4400 cores
- 50 TB main memory
- 30 A100 GPUs
- 100Gb network
- >2PB storage

- Operational in 2024

- Genomics, Proteomics
- Microscopy, Imaging
- Comp. Material Science
- Climate Modelling
- Earth Modeling
- Remote Sensing
- Large corpora in Digital Humanities
- Large Language Models
- HPC research
- …

Some Numbers

Institute for Computer Science

Total: ~300 MWh / Year

~100t CO2 (*)

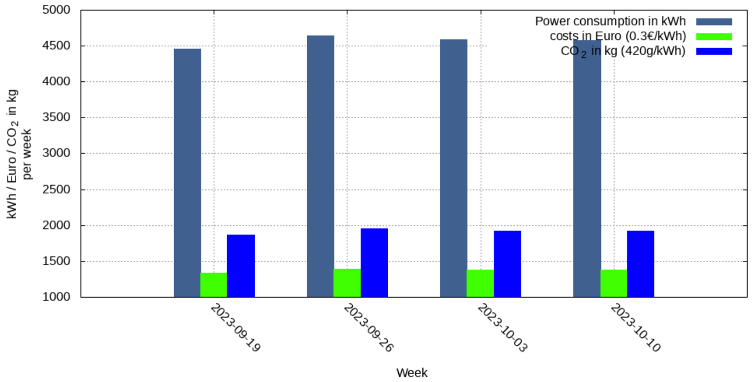
~90.000 Euro (*)

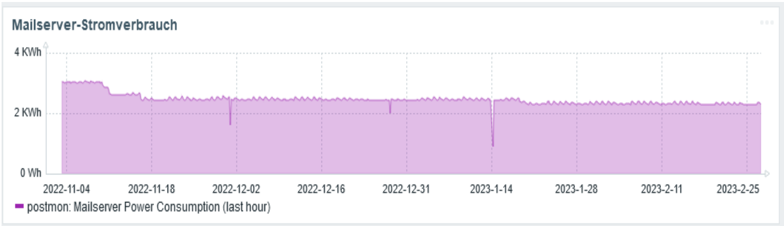
Computer and Media Center

Total: ~6.570 MWh / Year

~2.000t CO2 (*)

~1.970.000 Euro (*)





Saving Energy in Data Analysis

- Energy-efficient algorithms
- Carbon-aware scheduling
- Right-sizing of clusters
- Cluster consolidation
- Hardware switching: Clock-frequency, memory banks
- New hardware: GPUs, FPGA
- Energy-aware programming
- Energy-aware experiment planning in AI
  - Bootstrapping, hyperparameter, permutation testing, …

Who should be here

- Bachelor Informatik
- Ability to read English papers
- Knowledge in relevant Computer Science topics
  - Hardware, operating systems, distributed processing, algorithms, …
  - Optimization, heuristics, search spaces
- Willingness to work independently
  - Search suitable papers covering a topic, prepare presentations, write seminar thesis

How it will work

- Today: Presentation and choice of topics
  - If desired, we will group teams of 2 students
- 13.11.23: Send an outline of your topic (next slide)
- Before Christmas: Present your topic in 5min talk
- 31.01.24: Meet your advisor to discuss slides
- February: Present your topic in a Blockseminar
- 01.04.23: Write seminar thesis (10-15 pages)

The outline

- Topics will be rather abstract
- Find yourself a set of suitable papers
  - A specific focus is allowed and welcome
- Extract the most important information
- Structure into an outline of your thesis
  - Abstract, chapters, sections,
  - 1-2 sentences per section to describe the content
- Abstract
  - Roughly 20 lines – what is the topic, what will the thesis describe?
- Send us outline + references
  - Mark your top-3 references – those that most likely will form the basis of your work

The 5-min flash talk

- Focus on marketing – sell your topic to gain audience
  - What is the topic?
  - Why is it challenging?
  - Why is it cool?
  - What are important applications?
  - What will your talk be about?
- At most 5 slides
- Focus on figures & examples; omit details or algorithms

Presentation

- 20min presentation for 1-person groups, 30 for 2-person groups
- German or English
- Explain topic, methods, maybe experimental results
- Compare different approaches (if enough time)
- Aim: Your audience should understand what you say
- No need to cover the topic entirely – a clear focus is helpful

Teams

- If a topic is addressed by a team of two students, we expect
  - Read more papers
  - Have more topics in your outline and thesis
  - Write longer thesis
  - Presentations times remain the same – choose wisely

ToC

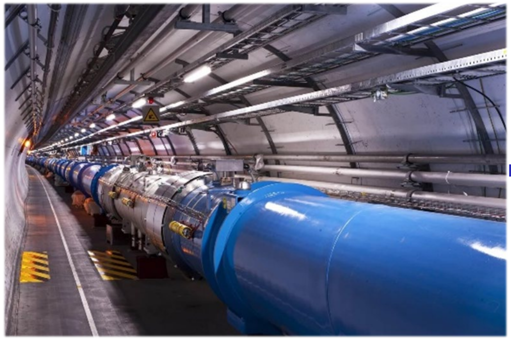
- Introduction
- Topics
- Assignment
- Hints on presenting your topic and writing your thesis

Topics

Topic	Advisor	Assigned to
Energy consumption of data science and regulations	FL	
Intel RAPL and beyond	UL	
Energy-efficient processing units: ARM, Intel, GPU, FPGA and beyond	UL	
Energy-efficient data / network transfer	FL	
Dynamic voltage and frequency scaling	UL	
Energy efficiency and programming languages	FL	
Energy efficient sorting	UL	
Energy-aware machine learning	UL	
Energy-aware query optimization	UL	
Predicting energy consumption of workflows	FL	
Energy-aware workflow scheduling	FL	
Carbon-aware scheduling	FL	

Energy consumption of data science and regulations (FL)

- Data science is not only about data processing but also collection
- Large laboratories, institutes, and universities have own observatories and server centers
- What is their energy consumption, and how sustainable is it
- What is planned to make it more sustainable
- Which governmental restrictions apply (Energieeffizienzgesetz)
- For example, the CERN



Intel RAPL and beyond (UL)

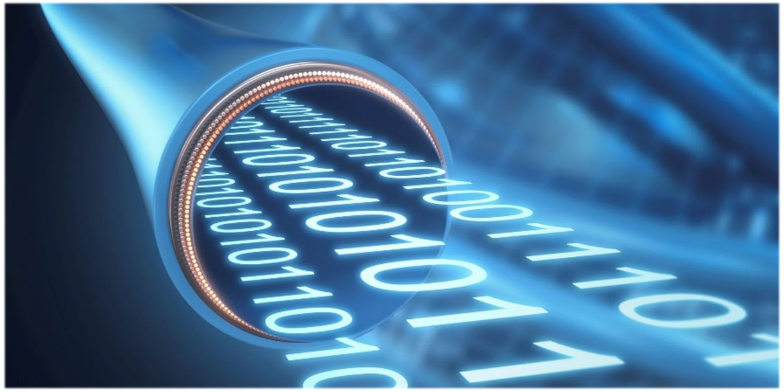
- Before optimizing for energy, we need to be able to measure energy consumption
- Hardware power meter or software-based solutions
- Intel RAPL – Framework for estimating energy consumption of programs using internal hardware counter
  - Running Average Power Limit Energy Reporting
- Accuracy? Similar tools on non-intel chips? Only CPU or also IO, network, memory access? Usage?
- Needs: Programming skills; operating systems
- Optional: Use practically and measure effects

Energy-efficient processing units (UL)

- Computers today have a large choice of programming units / accelerators: different CPUs, GPUs, FPGA, …
  - Degree of parallelization, complexity of operations. Static or programmable, memory access, …
  - Especially: Energy-efficiency (think mobile)
- What devices exist? Difference in energy consumption? Basis of reduction? Trade-offs?
- Needs: Low-level hardware knowledge
- Optional: Implement 1-2 problems on CPU / GPU or laptop / server / smartphone and measure

Energy-efficient data / network transfer (FL)

- Distributed data analysis are state-of-the-art (DAWs/Streaming)
  - Inter/intra cluster
  - For practical reasons (one machine is to small)
  - For legal reasons (raw data has to remain in place)
- Terabytes of data are moved between clusters or nodes
- Strategies to reduce network transfer (energy consumption)



Dynamic voltage and frequency scaling (UL)

- "Adjustment of power and speed settings on … processors, controller chips and peripheral devices to … maximize power saving when those resources are not needed."
  - https://www.techtarget.com/
- Energy consumption of many computer units are proportional to performance (speed)
  - Reducing frequency, bandwidth, etc. reduces energy consumption
  - At the price of lower speed
- Which units? Storage and memory? Network adapter? Proportionality? Detection of little use?
- Needs: Good understanding of computer architectures; some hardware knowledge (electrical engineering)
- Optional: Use practically and measure effects

Energy efficiency and programming languages (FL)

- The execution of code uses energy
- Consumption depends on
  - Algorithm
  - Implementation
  - Language & Compiler

- What to consider to reduce energy of your code
  - Best practices

binary-trees				
	Energy	Time	Ratio	Mb
(c) C	39.80	1125	0.035	131
(c) C++	41.23	1129	0.037	132
(c) Rust $\Downarrow_2$	49.07	1263	0.039	180
(c) Fortran $\Uparrow_1$	69.82	2112	0.033	133
(c) Ada $\Downarrow_1$	95.02	2822	0.034	197
(c) Ocaml $\downarrow_1$ $\Uparrow_2$	100.74	3525	0.029	148
(v) Java $\uparrow_1$ $\Downarrow_{16}$	111.84	3306	0.034	1120
(v) Lisp $\downarrow_3$ $\Downarrow_3$	149.55	10570	0.014	373
(v) Racket $\downarrow_4$ $\Downarrow_6$	155.81	11261	0.014	467
(i) Hack $\uparrow_2$ $\Downarrow_9$	156.71	4497	0.035	502
(v) C# $\downarrow_1$ $\Downarrow_1$	189.74	10797	0.018	427
(v) F# $\downarrow_3$ $\Downarrow_1$	207.13	15637	0.013	432
(c) Pascal $\downarrow_3$ $\Uparrow_5$	214.64	16079	0.013	256
(c) Chapel $\uparrow_5$ $\Uparrow_4$	237.29	7265	0.033	335
(v) Erlang $\uparrow_5$ $\Uparrow_1$	266.14	7327	0.036	433
(c) Haskell $\uparrow_2$ $\Downarrow_2$	270.15	11582	0.023	494
(i) Dart $\downarrow_1$ $\Uparrow_1$	290.27	17197	0.017	475
(i) JavaScript $\downarrow_2$ $\Downarrow_4$	312.14	21349	0.015	916
(i) TypeScript $\downarrow_2$ $\Downarrow_2$	315.10	21686	0.015	915
(c) Go $\uparrow_3$ $\Uparrow_{13}$	636.71	16292	0.039	228
(i) Jruby $\uparrow_2$ $\Downarrow_3$	720.53	19276	0.037	1671
(i) Ruby $\Uparrow_5$	855.12	26634	0.032	482
(i) PHP $\Uparrow_3$	1,397.51	42316	0.033	786
(i) Python $\Uparrow_{15}$	1,793.46	45003	0.040	275
(i) Lua $\downarrow_1$	2,452.04	209217	0.012	1961
(i) Perl $\uparrow_1$	3,542.20	96097	0.037	2148
(c) Swift	n.e.			

Pereira et al., "Energy Efficiency across Programming Languages"

VS

We expect **PRACTICAL EXPERIMENTS**!

Energy-efficient sorting (UL)

- Classical problem in energy-efficient algorithms
- Different sorting algorithms need different amounts of energy
  - Memory versus register,; random access versus sequential; avoiding branch miss-predictions; …
- Difference between in-memory and external sorting
- Joule Sort, Tritan Sort, Fawn Sort, ….
- What are the tricks? Effect strength? Data-type dependency? in-memory or external?
- Needs: Interest in low-level, hardware-dependent programming tricks; C
- Optional: Use practically and measure effects

Energy-aware machine learning (UL)

- Machine learning is the new prime energy consumer in IT
  - Exhaustive hyper-parameter sweeps; large linear optimization problems; permutation testing for significance; bootstrapping for uncertainty estimation; billion-parameter models trained on billion token corpora …
- Difference between model training and model use
- Many suggestions for becoming more energy-efficient
- How long to train? Are all training instances necessary? Model reuse? GPU versus CPU? Bayesian instead of grid-based HP optimization?
- Needs: Machine learning, optimization methods

Energy-aware query optimization (UL)

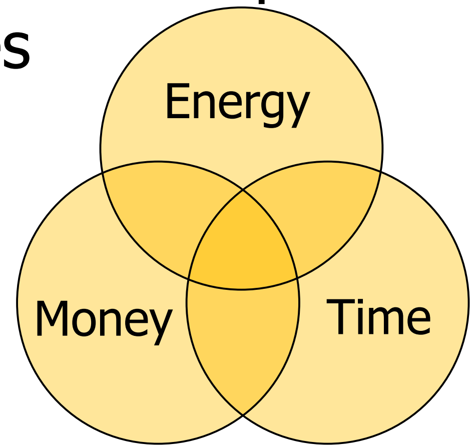
- Databases optimize queries before executing them
  - To find the probably fastest plan
- Optimal query plans might require more energy
  - And exhaustive optimization costs energy
- Trade-Off: Speed and energy consumption of query plans
  - Scans instead of index for sequential access; memory access instead of IO; plan reuse instead of new optimization; reduced optimality to reduce optimization time; …
- How to estimate energy consumption of a query plan? Of the optimizer? Which operators need more energy? Optimization for low energy consumption?
- Needs: relational databases, optimization algorithms
- Optional: Use practically and measure effects

Predicting energy consumption of workflows (FL)

- Workflows are used to gather information from data
- Long-running and large scale
- Execution in the Cloud and heterogeneous environments
- Knowledge of energy consumption: shift workload
  - To better fitting node
  - In time when more sustainable energy becomes available
- Look into prediction models of tasks and nodes

Energy-aware workflow scheduling (FL)

- Heterogeneous nodes consume different energy to process the same task
- Special hardware is optimized for specific tasks
- Make use of this knowledge and assign tasks to best fitting node
- Trade-off between runtime, energy consumption, and cost
- Practical: Use Wrench/WorkflowSim to compare HEFT with energy-aware scheduling strategies



Carbon-aware scheduling (FL)

- Many Workflows are long-running
- Access to different clusters
- Shift workflow temporal and spatial to reduce carbon emission
  - Temporal: time where more power is available/less cooling required
  - Spatial: Location with more power available/less cooling required