Green computing

Oleksiy M. Kozlov

The Exelixis Lab
Heidelberg Institute for Theoretical Studies
Germany

Český Krumlov – January 26, 2024
Green Computing: Challenges

- Extremely broad topic
  - From electric grid to CPU to software to users
  - Different cultures / mindsets / languages
  - “Divide-and-conquer” is problematic
- Many non-technical factors
  - Economics, politics, bureaucracy, psychology...
Agenda

• Energy monitoring
• Energy optimization
• Carbon-aware computing
Computing infrastructure survey

• What are you using for long analyses (>4h)?
  – Personal laptop or desktop
  – Dedicated desktop or server (lab-owned)
  – Shared server or cluster (institute-owned, external)
  – Cloud (e.g. AWS)
Our computing infrastructure

• Lab-owned: 4 desktops, 6 rack servers
• HITS institute cluster #1 (on premise)
• HITS institute cluster #2 (university)
• External resources (SCC Karlsruhe, LRZ Garching/Munich)
Motivation 2019: Climate crisis

- Back-of-the-napkin estimate:
  - 2 CPUs x 85W + peripherals + cooling >= 200W
  - 24h single-node job: 0.2 x 24 = 4.8 kWh
  - German energy mix 2019: ~400g CO2 / kWh
  - CO2 emissions: 4.8 x 0.4 = 1.92 kg

BMW X5 M50d: 190g/km
Software gets faster

480x speedup

EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences

4x speedup

RAXML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference

30x speedup

Multi-rate Poisson tree processes for single-locus species delimitation under maximum likelihood and Markov chain Monte Carlo

~1000x speedup
Hardware gets better

- **Green500**: Exponential growth in FLOPS/Watt
Top500 power trend
The Jevons paradox

- W. S. Jevons „The Coal Question“ (1865):
  - Improved efficiency
  - Increased consumption rate
  - Lower cost

a.k.a. “rebound effect” or “induced demand”

Energy monitoring
Energy monitoring: goals

• Measuring energy *accurately* is hard!

• Let’s be pragmatic
  – All measurements are wrong, but some are useful
  – Feasibility over Simplicity over Accuracy
  – Consistency important for comparability
Energy monitoring: use case #1

- General awareness
  - Impact vs. other areas, e.g. transportation
  - Averaged estimates are OK → W/core, CO2/kWh ...

Fig. 1: Average annual emissions in 2018 for an Australian and MPIA researcher in tCO₂e yr⁻¹, broken down by sources.

(Jahnke et al., Nat Astronomy 2020)
The analysis took **1,200,000 CPU-hours** on the SuperMUC-NG supercomputer (LRZ, Garching, Germany)

In 2021, LRZ energy consumption was **32,632,950 kWh** [LRZ1], and in total **2,308,500,000 CPU-hours** were allocated to user jobs [LRZ2]

On average, this corresponds to roughly **14 Wh per CPU-hour**, or **17,000 kWh** for the full analysis

This translates to ~**7,200 kg** of CO2 based on carbon intensity of the German electricity mix (0.425 kgCO2/kWh in 2021 [UBA])

This is roughly equivalent to **17 NY->London flights** (one-way) [Google]
Bottom-up estimation

http://calculator.green-algorithms.org/
NJ for Future?

(Kumar, 2022 MBE)

can raxml-ng handle 6 million genomes of SARS-CoV-2? #172

(MEGA)
Energy monitoring: use case #2

- Comparative analysis / benchmarking
  - Year-to-year, programs, parameters etc.
  - Actual measurements needed
  - Systematic under-/overestimation is OK
Energy measurement levels

- Building / Datacenter → smart meters
- Server room, rack → smart PDUs
- Node / Server
- CPU / GPU
- Job
- User
## Energy monitoring: Toolbox

<table>
<thead>
<tr>
<th></th>
<th>IPMI / DCMI</th>
<th>NV-SMI</th>
<th>ROcM-SMI</th>
<th>RAPL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Platform</strong></td>
<td>Servers</td>
<td>GPU (Nvidia)</td>
<td>GPU (AMD)</td>
<td>CPU (Intel, AMD)</td>
</tr>
<tr>
<td><strong>Scope</strong></td>
<td>Full system</td>
<td>GPU</td>
<td>GPU</td>
<td>CPU+DRAM</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td><strong>Energy</strong></td>
<td>×</td>
<td>×</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>very high</td>
</tr>
<tr>
<td><strong>Low latency</strong></td>
<td>×</td>
<td>depends?</td>
<td>???</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>×</td>
<td>???</td>
<td>???</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Non-root access</strong></td>
<td>×</td>
<td>✔</td>
<td>✔</td>
<td>✔ / ×</td>
</tr>
<tr>
<td><strong>Power limiting</strong></td>
<td>✔ / ×</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Energy monitoring: RAxML-NG

• New in RAxML-NG v1.0: energy usage report
  – Measured with Intel RAPL → CPU+DRAM only
  – Supported on Linux systems only
  – To disable, add: --extra energy-off

Elapsed time: 42846.287 seconds
Consumed energy: 162370.469 Wh (= 812 km in an electric car, or 4059 km with an e-scooter!)
Automatic parallelization tuning

Execution time (s)

Consumed energy (Wh)

Average power (W)

Number of threads

Number of threads

Number of threads

default: v1.0+ v0.9

-30%
Experiment Impact Tracker

- Energy&CO2 tracking library for Python
  - Code: https://github.com/Breakend/experiment-impact-tracker

```python
from experiment_impact_tracker.compute_tracker import ImpactTracker

experiment1 = tempfile.mkdtemp()
experiment2 = tempfile.mkdtemp()

with ImpactTracker(experiment1):
    do_something()

with ImpactTracker(experiment2):
    do_something_else()

$ generate-carbon-impact-statement experiment1_log_dir experiment2_log_dir
```
Energy monitoring in SLURM

- Plugins for RAPL, IPMI...
- Node power
- Job energy

```
[user@cascade-login ~]$ scontrol show node cascade-149
NodeName=cascade-149 Arch=x86_64 CoresPerSocket=20
    CPUAlloc=0 CPUTot=40 CPULoad=0.00
AvailableFeatures=cpu6230,ram96,rtx2080
ActiveFeatures=cpu6230,ram96,rtx2080
Gres=gpu:2,cpuonly:1
NodeAddr=cascade-149 NodeHostName=cascade-149 Version=20.11.7
OS=Linux 4.18.0-147.8.1.el8_1.x86_64 #1 SMP Thu Apr 9 13:49:54 UTC 2020
RealMemory=95000 AllocMem=0 FreeMem=92900 Sockets=1 Boards=1
MemSpecLimit=2048
State=IDLE ThreadsPerCore=2 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
Partitions=debug.p
BootTime=2021-06-09T11:20:43 SlurmdStartTime=2021-06-09T15:02:11
CfgTRES=cpu=40,mem=95000M,billing=40,gres/gpu=2
AllocTRES=
CapWatts=n/a
CurrentWatts=18 AveWatts=14
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Comment=(null)
```

```
[user@cascade-login ~]$ sacct -j 880244 -o ConsumedEnergy
ConsumedEnergy
------------------
  70.11K
  70.11K
  70.10K
```
Energy in SLURM: problems

- Poor visibility, no summary stats
- RAPL broken on Intel Haswell and later
  - Fixed on HITS clusters
  - Bug opened upstream: https://bugs.schedmd.com/show_bug.cgi?id=9956
- No support for NVIDIA GPUs (yet?)
  - But: AMD GPUs via ROCm-SMI library
Per-application energy

• What if nodes are shared?
• Perfect attribution problematic
• Heuristic: use CPU usage ratio (proc/sys)
Guerilla monitoring @ HITS

- **Group servers**
  - IPMI → telegraf → influxDB → Grafana
  - Resolution: 30 s
  - HowTo: [https://github.com/amkozlov/ipmi-grafana](https://github.com/amkozlov/ipmi-grafana)

- **HITS clusters**
  - IPMI → checkmk → CSV → influxDB → Grafana
  - Resolution: 1 min
Group servers power

Total energy consumed: 752 kWh

Idle energy saved: 149 kWh

Tom: 51.8 kWh
Jerry: 118 kWh
Deimos: 109 kWh
Phobos: 117 kWh
Cluster nodes power: aggregated

+ storage + network + cooling + PSU conversion losses
Energy optimization
Idle consumption

- Example: Intel i7-7800X, 6 cores, 64GB RAM
  - sleep: 5W, idle: 55W, under load: 150W
  - 50% utilization → 25% savings (219 kWh/a)
  - 30% utilization → 42% savings (306 kWh/a)
- Example: Xeon Platinum 8260, 48 cores, 764GB RAM
  - standby: 20W, idle: 120 W, under load: 500 W
  - 50% utilization → 16% savings (438 kWh/a)
  - 30% utilization → 30% savings (613 kWh/a)
Sleep-on-Idle

• Desktops
  - Suspend-on-idle + Wake-on-LAN
  - Used by several groups, no centralized solution yet

• Rack servers
  - No suspend to RAM :(
  - PowerOff-on-idle + PowerOn-over-IPMI
  - https://github.com/amkozlov/idle-sleep
PowerOff-on-Idle

- Idle detection cron job
  - No active sessions + CPU utilization < 0.5 → idle
  - Idle since 1 hour → poweroff
  - Can be temporarily disabled:
    ```
    deimos$ ecosleep disable 12h
    Server will not be powered off until: Di 16. Mai 10:13:34 CEST 2023
    ```

- PowerOn via SSH
  ```
  laptop$ alias | grep ecowake
  alias ecowake='ssh kozlovay@XXXX.h-its.org sudo /hits/fast/cme/ecosleep/wakeup.sh'
  
  laptop$ ecowake deimos
  kozlovay@XXX.h-its.org's password:
  Chassis Power Control: Up/On
  waiting for deimos .................................................................
  Server is back online!
  ```
Idle power savings 2023

- 1600 kWh (16%)
- -11% vs. 2022
Problems / Improvements

- screen/tmux sessions lost
  - tmux-resurrect might help
- Boot delay 1-2 min.
Power scaling
Power scaling on CPU and GPU

- Widely available: Intel/AMD/NVIDIA
- Power and/or frequency limits
- Typical range: 50% - 100% TDP
- Easy-to-use, transparent to workload

NVIDIA GeForce RTX 2080 SUPER

```bash
$ nvidia-smi -q -d POWER,CLOCK
Power Management                  : Supported
Power Draw                        : 4.24 W
Power Limit                       : 250.00 W
Default Power Limit               : 250.00 W
Enforced Power Limit              : 250.00 W
Min Power Limit                   : 125.00 W
Max Power Limit                   : 250.00 W
```

```bash
$ sudo nvidia-smi -pl 200
Power limit for GPU 00000000:17:00.0 was set to 200.00 W from 300.00 W.
```

Intel Xeon Platinum 8260

```bash
$ sudo cpupower frequency-info
hardware limits: 1000 MHz - 3.90 GHz
available cpufreq governors: performance powersave
current policy: frequency should be within 1000 MHz and 3.90 GHz.
```

AMD EPYC 7452

```bash
$ sudo cpupower frequency-info
hardware limits: 1.50 GHz - 2.35 GHz
available frequency steps: 2.35 GHz, 2.00 GHz, 1.50 GHz
```

```bash
$ sudo cpupower frequency-set -u 2000000
Setting cpu: 0
...```
Energy efficiency “sweet spot”

(RAxML-NG 1.1, 2x Intel Xeon Platinum 8260, 48T)
Workload variation
Motivation 2022: Energy crisis

Reduce electricity consumption

THE COMMISSION PROPOSES:

A target for Member States to reduce overall electricity demand by at least 10%

Demand during peak hours

An obligation for Member States to reduce demand during peak price hours by at least 5%

By reducing electricity demand by 5% at peak times, we reduce gas use for power by around 4% over the winter and reduce pressure on prices

Source: https://ec.europa.eu/commission/presscorner/detail/en/fs_22_5491
HPC community reaction

(biased subjective perception, limited sample size)
“Christmas experiment”

- **23.12.2022** → Apply power throttling
  - Haswell cluster (URZ): CPU 2000 MHz
  - Cascade cluster (HITS): CPU 90 W / GPU 175 W
- **16.01.2023** → Back to normal power
- **16.01. – 29.01.** → Baseline data collection
Energy consumption: HITS campus

Monthly: 2022/23 vs. 2021/22

December: -8%
January: -15%

Daily: 2022/23 vs. 2021/22 (kWh)

cluster throttled
# Energy consumption: Clusters

<table>
<thead>
<tr>
<th></th>
<th>Avg. node power (excl. idle)</th>
<th>Energy saved</th>
<th>Performance reduction (min ... max)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Regular 16.01 – 29.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Haswell / URZ</td>
<td>150 W</td>
<td>5110 kWh</td>
<td>-12 % ... -32 %</td>
</tr>
<tr>
<td>Cascade / HITS</td>
<td>294 W</td>
<td>4791 kWh</td>
<td>-10 % ... -25 %</td>
</tr>
<tr>
<td></td>
<td>Reduced 23.12 – 15.01*</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>220 W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Diff.</td>
<td>-33 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-25 %</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* cascade: 07.01-11.01 excluded due to storage failure

Estimated total savings (3 weeks):

⚡ 10,000 kWh = 🏡 5-7 years = 🚗 50,000 km
Performance vs. power reduction

- performance loss is sublinear w.r.t. power
- BUT: for many workloads, “free lunch” is small: 2000 MHz below efficiency sweet spot?
Searching for the sweet spot

Max frequency (MHz)

<table>
<thead>
<tr>
<th></th>
<th>2200</th>
<th></th>
<th></th>
<th></th>
<th>2400</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GROMACS−16</td>
<td>−30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAXNG_AA−8</td>
<td>−20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLH−16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLH−8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Difference (%)

-30 -20 -10 0 10 20 30

- performance
- power
Carbon-aware computing
Real-time energy mix

Carbon intensity in the last 24 hours

Origin of electricity in the last 24 hours

Electricity prices in the last 24 hours

https://app.electricitymaps.com
Signal: Local, marginal, CO2, price

https://www.stromgedacht.de/

https://tibber.com
Compute with cleaner energy

- Proof-of-concept implementation: https://github.com/amkozlov/eco-freq
Power scaling: techniques

- **RAPL / DVFS**
  - Dynamic power / frequency limits → \(~50\% - 100\%\) TDP
  - Supported by most CPUs/GPUs (Intel, AMD, NVIDIA)

- **Utilization capping**
  - e.g. Linux cgroup

- **Adaptive parallelization**
  - Adjust # threads / MPI ranks

- **Freeze / suspend / turn off nodes**
RAPL: advantages

- Transparent to the workload
  - No profiling, recompilation etc.
  - Long jobs are fine (no interruption / restart)
- Also works without job queue / scheduler
- No generation forecast needed
  - But can be used if available
**EcoFreq: Demo**

$ sudo ./ecofreq.py

```
2021-06-11T23:14:18            380           223.000              NA         398.650    358785.000          37.874
...  
2021-06-12T09:44:48            262           275.750              NA         529.639    476675.000          34.632
...  
2021-06-12T13:44:59            133           330.000              NA         594.097    534687.500          19.778
```

$ ./ecostat.py

```
EcoStat v0.0.1
Loading data from log file: /var/log/ecofreq.log

Time interval: 2021-05-18 - 2021-06-11
Duration active: 23 days, 23:15:57
Duration inactive: 17:06:18
CO2 intensity range [g/kWh]: 108 - 449
CO2 intensity mean [g/kWh]: 284
Energy consumed [J]: 657629645.0
Energy consumed [kWh]: 182.675
CO2 emitted [kg]: 51.701283
```

$ sudo ./ecostat.py

```
$ ./ecostat.py
EcoStat v0.0.1
Loading data from log file: /var/log/ecofreq.log

Time interval: 2021-05-18 - 2021-06-11
Duration active: 23 days, 23:15:57
Duration inactive: 17:06:18
CO2 intensity range [g/kWh]: 108 - 449
CO2 intensity mean [g/kWh]: 284
Energy consumed [J]: 657629645.0
Energy consumed [kWh]: 182.675
CO2 emitted [kg]: 51.701283
```

$ ./ecostat.py

```
$ ./ecostat.py
EcoStat v0.0.1
Loading data from log file: /var/log/ecofreq.log

Time interval: 2021-05-18 - 2021-06-11
Duration active: 23 days, 23:15:57
Duration inactive: 17:06:18
CO2 intensity range [g/kWh]: 108 - 449
CO2 intensity mean [g/kWh]: 284
Energy consumed [J]: 657629645.0
Energy consumed [kWh]: 182.675
CO2 emitted [kg]: 51.701283
```

$ ./ecostat.py

```
$ ./ecostat.py
EcoStat v0.0.1
Loading data from log file: /var/log/ecofreq.log

Time interval: 2021-05-18 - 2021-06-11
Duration active: 23 days, 23:15:57
Duration inactive: 17:06:18
CO2 intensity range [g/kWh]: 108 - 449
CO2 intensity mean [g/kWh]: 284
Energy consumed [J]: 657629645.0
Energy consumed [kWh]: 182.675
CO2 emitted [kg]: 51.701283
```

$ ./ecorun.py -p linear raxml-ng

```
[...]

time_s: 882.708
pwr_avg_w: 553.422
energy_j: 488510.0
energy_kwh: 0.136
co2_g: 51.112
```

$ ./ecorun.py -p linear raxml-ng

```
[...]

time_s: 882.708
pwr_avg_w: 553.422
energy_j: 488510.0
energy_kwh: 0.136
co2_g: 51.112
```

$ ./ecorun.py -p linear raxml-ng

```
[...]

time_s: 882.708
pwr_avg_w: 553.422
energy_j: 488510.0
energy_kwh: 0.136
co2_g: 51.112
```
EcoFreq: Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>EcoFreq</th>
<th>Diff. %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time [s]</td>
<td>58502</td>
<td>62257</td>
<td>+6.4 %</td>
</tr>
<tr>
<td>Energy [kWh]</td>
<td>9.533</td>
<td>9.082</td>
<td>-4.7 %</td>
</tr>
<tr>
<td>CO2-to-solution [g]</td>
<td>2590</td>
<td>2307</td>
<td>-10.9 %</td>
</tr>
<tr>
<td>CO2-per-hour [g] (100% utilization)</td>
<td>153</td>
<td>133</td>
<td>-13.1 %</td>
</tr>
</tbody>
</table>

Germany, June 11-12, 2021
Further topics

- Cooling
- Heat re-use
- UPS
- Datacenter location
- Embodied carbon
- E-waste
- ...

https://arxiv.org/abs/2311.01169
Take-home messages

• Familiarize yourself with energy
  – measured in kWh, not “average households”

• Look for absolute consumption numbers
  – Not hypothetical “savings”

• Don’t wait for management or IT
  – Find motivated colleagues, build horizontal links