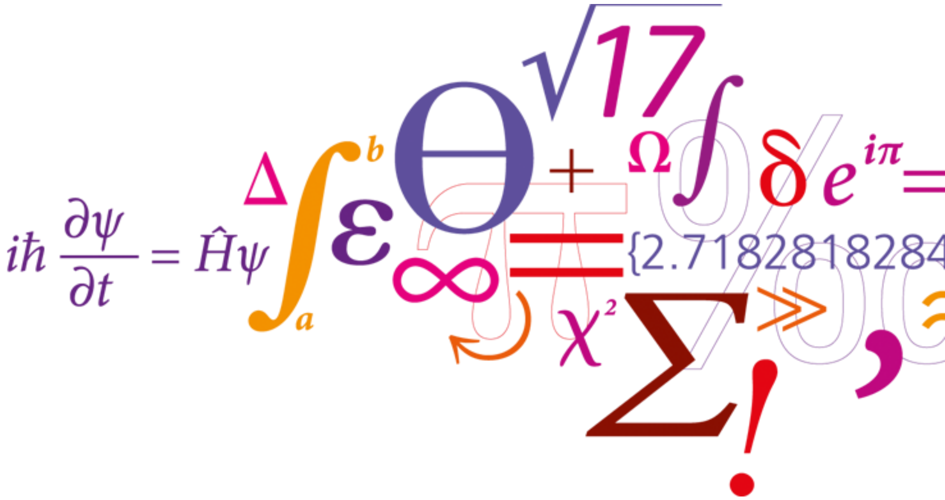
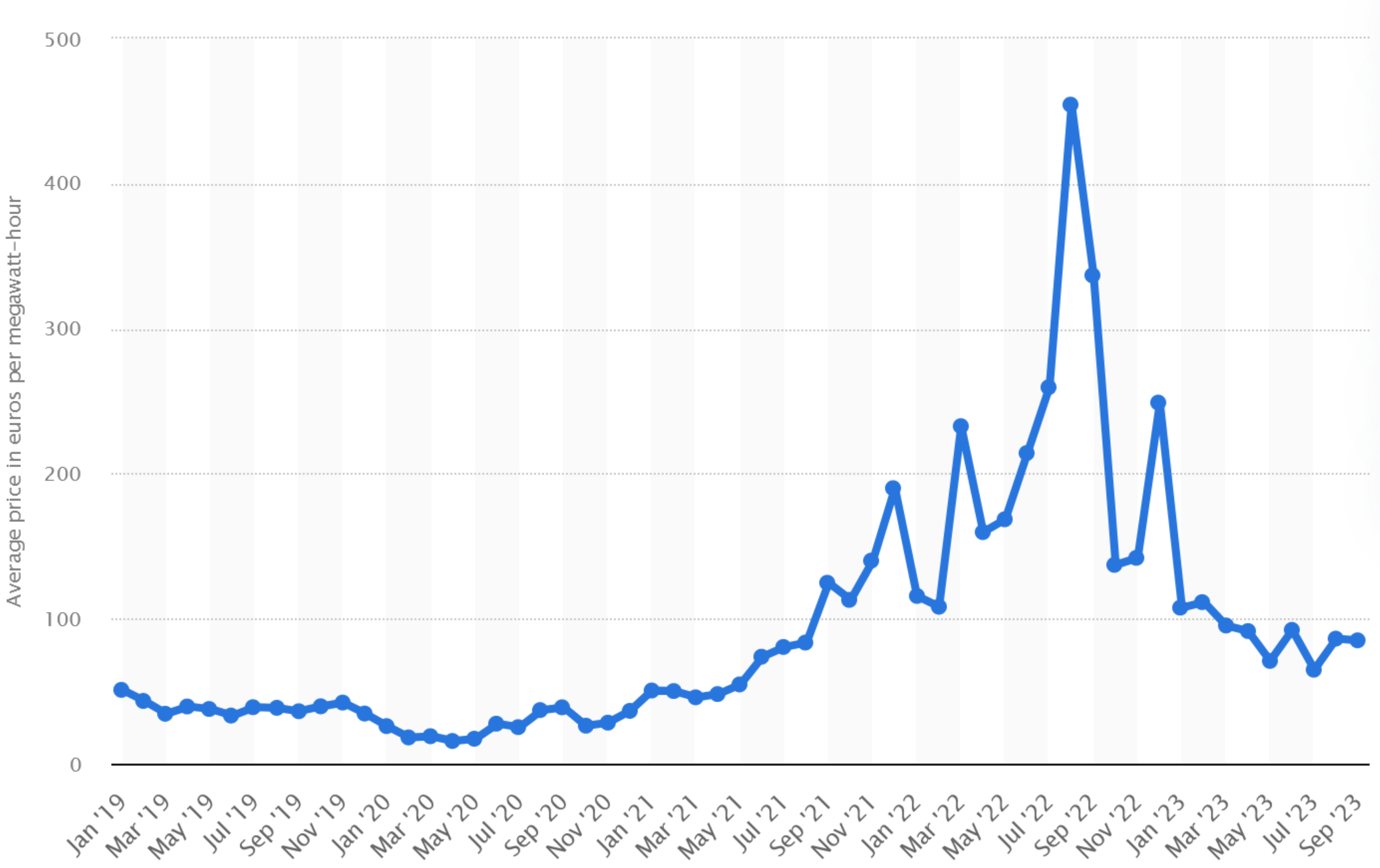
# Power Saving with Slurm

Ole Holm Nielsen, Ph.D.
DTU Fysik
Technical University of Denmark

Email: Ole.H.Nielsen@fysik.dtu.dk
Wiki: https://wiki.fysik.dtu.dk/Niflheim_system/
Tools: https://github.com/OleHolmNielsen/Slurm_tools



# Average monthly electricity wholesale price in Denmark (Euros per MWh)



April 2020: 15 €

August 2022: 454 €

September 2023: 85 €

Statista 2023

# Energy saving has become a political priority

- Even though electricity prices in Europe have come down from the peak in 2022, prices are still high, and our governments put a high priority on saving electricity.
- Non-fossil energy sources such as wind, hydroelectric, solar power, and nuclear have increased political priority in Europe.
- Electricity costs have become highly visible in public and private organizations.
- HPC centers are feeling a pressure to reduce the consumption of electricity.

# HPC supercomputers: The role of Slurm

- Historically Slurm was an acronym standing for *“Simple Linux Utility for Resource Management”*.

- Development started in 2002 at Lawrence Livermore National Laboratory as a resource manager for Linux clusters.
- Sophisticated scheduling plugins added in 2008.
- About 500,000 lines of C code today (plus test suite and docs).

- Used on many of the world's largest computers (for example, *LUMI*).
- Active global development community.

# Saving electricity by turning idle nodes off

- Many HPC centers will have some fraction of compute nodes which are **idle** (i.e., not running jobs) from time to time.

- Slurm's *slurmctld* controller can **suspend** (turn off) idle compute nodes, both in **on-premise** clusters and in **clouds** as described in the Slurm *Power Saving Guide*: https://slurm.schedmd.com/power_save.html

- The *slurm.conf* **SuspendProgram** and the **SuspendTime** partition parameters determine how and when nodes get turned off. Unfortunately, no actual methods for turning nodes off and on are provided in Slurm's code examples or documentation.

- **Note**: Some network fabrics or storage systems may experience issues if nodes are turned off:
  - CRAY Slingshot (depending on software version).
  - InfiniBand and Omni-Path *Fabric Manager* nodes.
  - GPFS *Quorum nodes*.

# This project: Slurm power saving scripts

- Documentation of Slurm power saving is in this Wiki page: https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#configuring-slurm-conf-for-power-saving

- Our Slurm power saving tools are available on *GitHub*: https://github.com/OleHolmNielsen/Slurm_tools/tree/master/power_save

- The basic concepts are:
  1. Use "*IPMI*" commands to control the compute node's power through the ***Baseboard Management Controller*** (BMC).
  2. The *slurmctld* server must have IP connectivity to the BMC's IP network (which may be a closed management network), possibly by using a dedicated NIC.
  3. Any firewall (if present) must permit *IPMI* traffic on port 623/udp.
  4. We use the **FreeIPMI** `ipmi-power` command to issue power on/off commands to the BMC. Note: The *slurmctld* will fork suspend/resume tasks running as the unprivileged user *slurm*.

# Setting up remote IPMI logins

- Define IPMI login variables *IPMI_USER* and *IPMI_PASSWORD*
  in the *slurm* user's *.bashrc* file (for example) on the *slurmctld* server.
  Alternative places to store login variables could be implemented.

- Set up all compute nodes for IPMI remote access via LAN using our `ipmi_setup` script.
  This essentially executes these commands:

  ```
  ipmitool user set name $userno $IPMI_USER
  ipmitool user set password $userno $IPMI_PASSWORD
  ```

- We assume that the BMC's DNS name is the node's DNS name plus some suffix.
  For example, node *a123* has BMC address *a123b.*

  Other one-to-one mappings of node-to-BMC DNS names could be implemented.

# Testing the power_ipmi script

- First test the IPMI scripts by querying some nodes (as user *slurm*) on the *slurmctld* server:

```
[slurm@ctld ~]$ power_ipmi -q d004,d005,c190
----------------
d004b,d005b
----------------
 on
----------------
c190b
----------------
 off
```

- Make a test of *suspending* (i.e., *power off*) a drained node.
  Wait for some minutes, then *resume* (i.e., *power on*) the node again:

```
[slurm@ctld ~]$ power_ipmi -s <nodename>
[slurm@ctld ~]$ power_ipmi -r <nodename>
```

- The power saving script will log IPMI power actions in a file in the *slurmctld* log directory:
  `/var/log/slurm/power_ipmi.log`

# Configuring suspend/resume in slurm.conf

- Define suspend and resume scripts:

```
ResumeProgram=/usr/local/bin/noderesume
ResumeFailProgram=/usr/local/bin/nodefailresume
SuspendProgram=/usr/local/bin/nodesuspend
```

- Define suspend and resume parameters:

```
PrivateData=cloud            # Bug 14270: Make Down nodes visible to sinfo. Resolved in 23.02.
ResumeTimeout=600
ResumeRate=30
SuspendTimeout=120
```

- Define an additional **`power_ipmi`** node feature for nodes managed by IPMI in the suspend/resume scripts:

```
NodeName=node[001-100] Feature=…,power_ipmi
```

# Partition configuration in slurm.conf

- Certain partitions or nodes may be **excluded from power saving**, if desired:
  ```
  SuspendExcParts=xeon40
  SuspendExcNodes=nodes[001-002]
  ```

- Only **partitions** for which you actually want power saving should have the `SuspendTime` flag:
  ```
  PartitionName=my_partition <…> SuspendTime=3600
  ```

- **Notes**:
  - Do **NOT** define a **global** `SuspendTime` flag! Keep the default `SuspendTime=Infinite`.
  - Choose a reasonable `SuspendTime` value so that nodes do not power down too soon:
    - You want new jobs to be able to start without delay.
    - Do not waste CPU time due to frequent reboots.

- **Start power saving**: Setting `SuspendTime` to anything but the default value of `INFINITE` will start Slurm power saving immediately as soon as *slurmctld* is reconfigured (`scontrol reconfig`).

# Operational issues: Our experiences

- Occasionally, nodes will be set to *Down* or *Drained* states for various abnormal reasons:
  - Hardware failure.
  - Disk full.
  - Networking error.
  - *Node Health Check* (LBNL NHC) detects an error.
  - OS or software errors.
  - Scheduled OS or application software updates.
  - Node OS reinstall.

- **Beware**: The Slurm *power_save* module prior to 23.02 didn't care about nodes in *Down* or *Drained* states!!
  - After `SuspendTime` Slurm will power down the node, and later resume it when needed by a job.
  - **Tedious workaround**: You can use `SuspendExcNodes` in *slurm.conf* and reconfigure. ☹

# Operational issues (cont’d)

- Nodes that are powered off by Slurm (having *State=Down*) can be difficult to manage:
  - Can’t have OS updates installed on the local hard disk.
  - Can’t update node firmware with OS-based CLI tools.

# Operational issues (cont'd)

- IPMI remote commands may be fail when:
  - A node got its motherboard replaced, including a new BMC with default settings.
  - The BMC got reset to default settings.
  - The BMC hardware has failed physically.
  - A multi-node chassis has failed causing BMCs to become inaccessible.
  - The BMC IP subnet has failed (switches or cables, for example).

- Checks to do:
  - Check the log file `/var/log/slurm/power_ipmi.log`
  - Ping the BMC's IP address.
  - Check the BMC's physical network connection.
  - Query the BMC using the `power_ipmi -q <nodename>` command, or `ipmitool`, or FreeIPMI tools.

# New in Slurm 23.02: SuspendExcStates

- Thanks to the fix in bug 15184 we have in 23.02 a new parameter for **excluding certain node states from suspension**, see the *slurm.conf* manual page:

- **SuspendExcStates**:
  - Specifies node states that are **not** to be powered down automatically.
    Valid states include CLOUD, DOWN, DRAIN, DYNAMIC_FUTURE, DYNAMIC_NORM, FAIL, INVALID_REG, MAINTENANCE, NOT_RESPONDING, PERFCTRS, PLANNED, and RESERVED.

- The **SuspendExcStates** parameter thereby enables us to automatically exclude nodes which are in the **DOWN** or **DRAIN** as well as other states!

  The following states should probably be excluded:

  ```
  SuspendExcStates=down,drain,fail
  ```

# Hardware stability with power saving

- Some computers may not tolerate frequent power off-then-on events very well:
  - Thermal stresses due to temperature changes may affect electronics and solder joints.
  - Latent errors in DIMM modules or other chips may only surface following a power cycle (they would occur sooner or later anyhow).
- Our experiences:
  - We configure `SuspendTime=3600` to avoid too frequent power cycles.
  - Hardware errors following power cycles seem to depend on the server model!
  - Most servers have the **same failure rate** as before we implemented power saving, but one specific "cloud" server model in our cluster fails quite often within hours or days of a power cycle.
    The failures are mostly due to a defective motherboard, but also DIMM and CPU failures occur.
  - Make sure that all **firmwares are updated** to the latest level (BIOS, BMC, CPLD, etc.) so that any fixes related to hardware are applied!
- Recommendation: Maintain a **valid service contract** to get broken nodes fixed.

# Monitoring compute node power

- An **idle node** may consume ~250W or about 30% of max power, depending on CPU model, GPU model (if any), and the thermal environment.
  Many BMCs include a power monitoring feature in a Web GUI or using CLI commands.

- **Node power may be monitored using Slurm** as described in https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_configuration/#power-monitoring-and-management using the **acct_gather_energy/ipmi** plugin with **DCMI**. Configure *slurm.conf* with:

```
AcctGatherEnergyType=acct_gather_energy/ipmi
AcctGatherNodeFreq=30
```

- Notes:
  - A bug in *slurmd* was fixed in bug 17639 and you must use Slurm 23.02.7 (or later).
  - Some vendor BMCs unfortunately do not offer the *IPMI* **DCMI** power statistics, see bug 17704 ☹
  - An alternative to *IPMI* is "*RAPL*" which provides CPU+DIMM monitoring only.

- Slurm can now report the current node power:

```
$ scontrol show node n123
CurrentWatts=641 AveWatts=480
```

- The `showpower` script linked in the above page can give useful partition or cluster summaries.

# Quantifying power savings

- Slurm’s *sreport* tool reports the percentage of *Down* (i.e., suspended) nodes:

```
$ sreport cluster utilization Start=0715 End=now -t percent
--------------------------------------------------------------------------------
Cluster Utilization 2023-07-15T00:00:00 - 2023-07-17T10:59:59
Usage reported in Percentage of Total
--------------------------------------------------------------------------------
Cluster  Allocated      Down PLND Dow     Idle   Planned   Reported
--------- ---------- --------- -------- -------- --------- ----------
niflheim     87.18%     8.59%    0.00%    0.00%     4.23%    100.00%
```

- In our cluster 8.59% corresponds to 58 nodes.
- Each suspended node has saved 250 W times 59 hours (the report period) for a **total savings of 855 kWh**.
- **Note**: The *Planned Downtime* (“`PLND Dow`”) ought to include all suspended nodes, but currently (up to 23.02) omits on-premise nodes and only reports cloud nodes.
  This will be fixed in Slurm 23.11 according to bug 17689.

# Conclusions

- Prices of electricity have risen sharply in Europe during 2022, and are not yet down to the pre-2022 levels.

- Significant pressure from management to **save electricity**, and a focus at the governmental level.

- Slurm’s *power_save* module enables **automatic shutdown of nodes** when there are no pending jobs.

- From Slurm 23.02 the bugs and shortcomings in the *power_save* module have been fixed so that power up/down operations have become convenient to use.

- Idle nodes may consume about 30% of the maximum node power, so **substantial electricity bill savings may be realized by power saving!**
- Cluster **power usage** can be monitored by Slurm.
- **Power savings** can be calculated from the power of idle nodes and using the `sreport` command to report the percentage of *Powered down* nodes in the cluster.