

Enabling Carbon Awareness through an Operating-System Daemon (carbond)

September 29, 2023

Herbsttreffen der Fachgruppe Betriebssysteme, Bamberg

Benedict Herzog², Andreas Schmidt¹, Gregory Stock¹, Robin Ohs¹, Luis Gerhorst^{2,3}, Timo Hönig²

¹Saarland Informatics Campus (SIC)

²**Ruhr-Universität Bochum (RUB)**

³Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)





GE-645 system running Multics (~1967) [1]

Management and distribution of resources was always domain of the operating system

► e.g., **Multiplexed** Information and Computing Service (Multics)



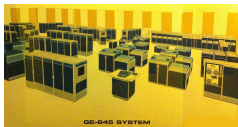
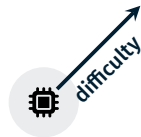
Andreas Schmidt, Gregory Stock, Robin Ohs, Luis Gerhorst, Benedict Herzog, Timo Hönig
"carbond: An Operating-System Daemon for Carbon Awareness."

In: *Proceedings of the 2nd Workshop on Sustainable Computer Systems (HotCarbon '23)*.

OS Resource Management

OS needs to attribute:

- hardware usage



~1960s

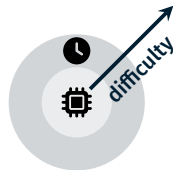


2023

OS Resource Management

OS needs to attribute:

- hardware usage
- time sharing



~1960s

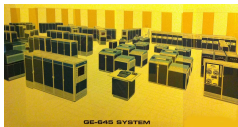
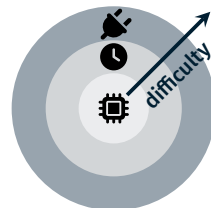


2023

OS Resource Management

OS needs to attribute:

- hardware usage
- time sharing
- energy consumption



~1960s

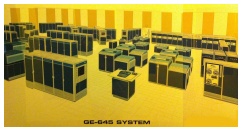
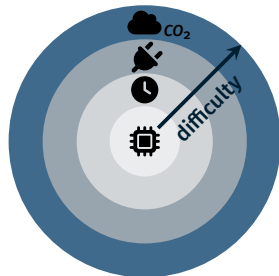


2023

OS Resource Management

OS needs to attribute:

- hardware usage
- time sharing
- energy consumption
- **now: carbon emissions**
 - ▶ operating-system daemon: carbond



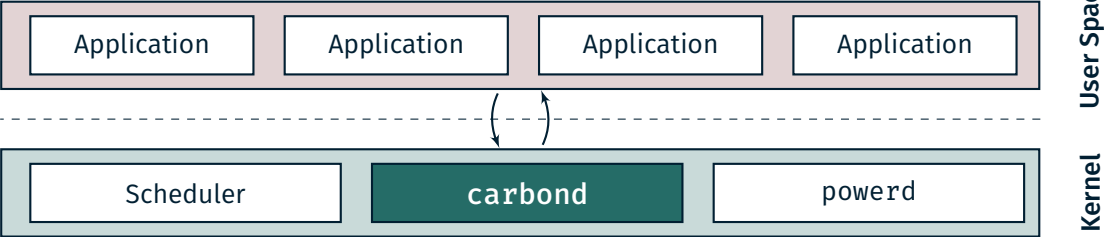
~1960s



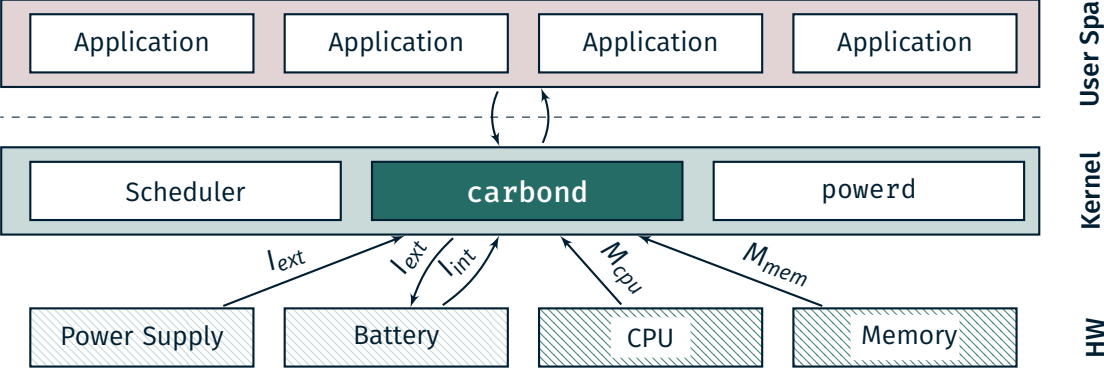
2023



carbond System Architecture



carbond System Architecture



/var/carbond/
└── emissions

OPERATIONAL EMISSIONS

$$O = E \cdot I$$

E: total energy consumption [kWh]

I: carbon intensity [$\frac{gCO_2}{kWh}$]

EMBODIED EMISSIONS

$$M = TE \cdot TS \cdot RS$$

TE: total embodied emissions [gCO_2]

TS: time share (of total lifespan)

RS: resource share

`/var/carbond/`
└── operational
└── embodied

OPERATIONAL EMISSIONS

$$O = E \cdot I$$

E: total energy consumption [kWh]

I: carbon intensity [$\frac{gCO_2}{kWh}$]

EMBODIED EMISSIONS

$$M = TE \cdot TS \cdot RS$$

TE: total embodied emissions [gCO_2]

TS: time share (of total lifespan)

RS: resource share

/var/carbond/
├── operational
└── embodied

OPERATIONAL EMISSIONS

$$O = E \cdot I$$

E: total energy consumption [kWh]

I: carbon intensity [$\frac{gCO_2}{kWh}$]

EMBODIED EMISSIONS

$$M = TE \cdot TS \cdot RS$$

TE: total embodied emissions [gCO_2]

TS: time share (of total lifespan)

RS: resource share

/var/carbon/
├── operational
└── embodied

OPERATIONAL EMISSIONS

$$O = E \cdot I$$

E: total energy consumption [kWh]

I: carbon intensity [$\frac{gCO_2}{kWh}$]

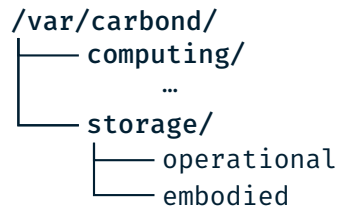
EMBODIED EMISSIONS

$$M = TE \cdot TS \cdot RS$$

TE: total embodied emissions [gCO_2]

TS: time share (of total lifespan)

RS: resource share



OPERATIONAL EMISSIONS

$$O = E \cdot I$$

E: total energy consumption [kWh]

I: carbon intensity [$\frac{gCO_2}{kWh}$]

EMBODIED EMISSIONS

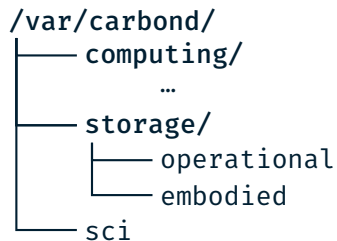
$$M = TE \cdot TS \cdot RS$$

TE: total embodied emissions [gCO_2]

TS: time share (of total lifespan)

RS: resource share

$$SCI = \frac{O+M}{R}$$



OPERATIONAL EMISSIONS

$$O = E \cdot I$$

E: total energy consumption [kWh]

I: carbon intensity [$\frac{gCO_2}{kWh}$]

EMBODIED EMISSIONS

$$M = TE \cdot TS \cdot RS$$

TE: total embodied emissions [gCO_2]

TS: time share (of total lifespan)

RS: resource share

Programming Library

- ▶ `resourcegauge-rs`

User-Space Tools

- ▶ `cperf`
- ▶ Carbon Flame Graphs



Programming Library

- ▶ resourcegauge-rs¹

User-Space Tools

- ▶ cperf
- ▶ Carbon Flame Graphs



```
1 #[resourcegauge(max_energy="10joule")]
2 fn compute () -> T {
3     // [...]
4 }
```

¹Schmidt et al.; ResourceGauge: Enabling Resource-Aware Software Components; OSPERT 2023.

Programming Library

- ▶ resourcegauge-rs¹

User-Space Tools

- ▶ cperf
- ▶ Carbon Flame Graphs



```
1 #[resourcegauge(max_carbon="10g")]
2 fn compute () -> T {
3     // [...]
4 }
```

¹Schmidt et al.; ResourceGauge: Enabling Resource-Aware Software Components; OSPERT 2023.

Programming Library

- ▶ resourcegauge-rs

User-Space Tools

- ▶ cperf
- ▶ Carbon Flame Graphs



```
1 $> perf stat -e "power/energy-pkg/" sleep 2
2 Performance counter stats for 'system wide':
3
4 5.62 Joules power/energy-pkg/
```

Programming Library

- ▶ resourcegauge-rs

User-Space Tools

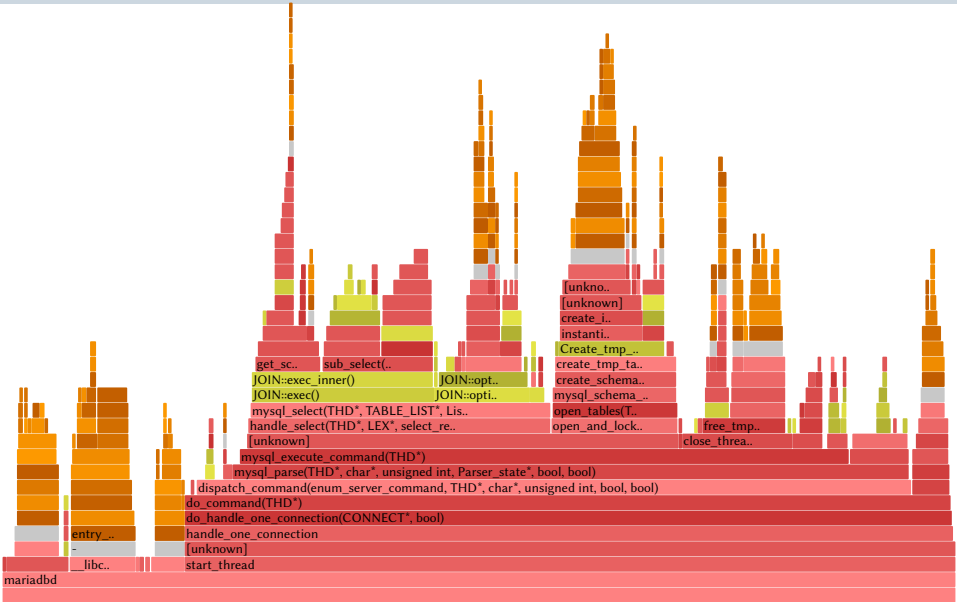


- ▶ cperf

- ▶ Carbon Flame Graphs

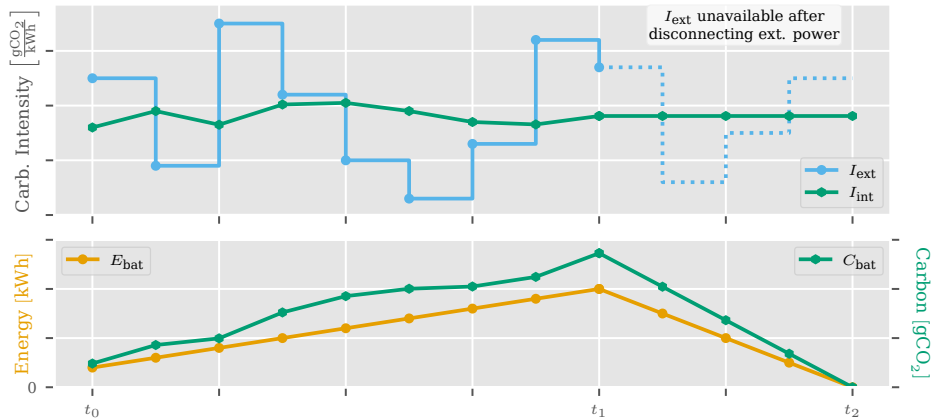
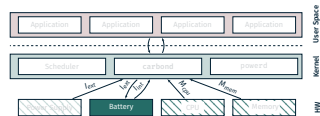
```
1 $> perf stat -e "carbon/cpu/" sleep 2
2 Performance counter stats for 'carbon cpu':
3
4 1.32 g CO2 carbon/cpu/ (0.3 embodied; 1.02 operational)
```


Carbon Flame Graphs



Time

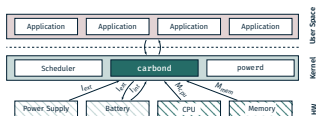
Battery Model



Conclusion

Enabling Carbon Awareness through an Operating-System Daemon (carbond):

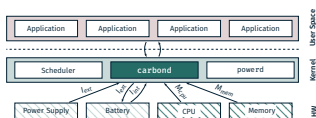
- collecting + providing carbon data
- located within the operating system
- currently implementing
 - carbond
 - cperf
 - Energy/Carbon Flame Graphs



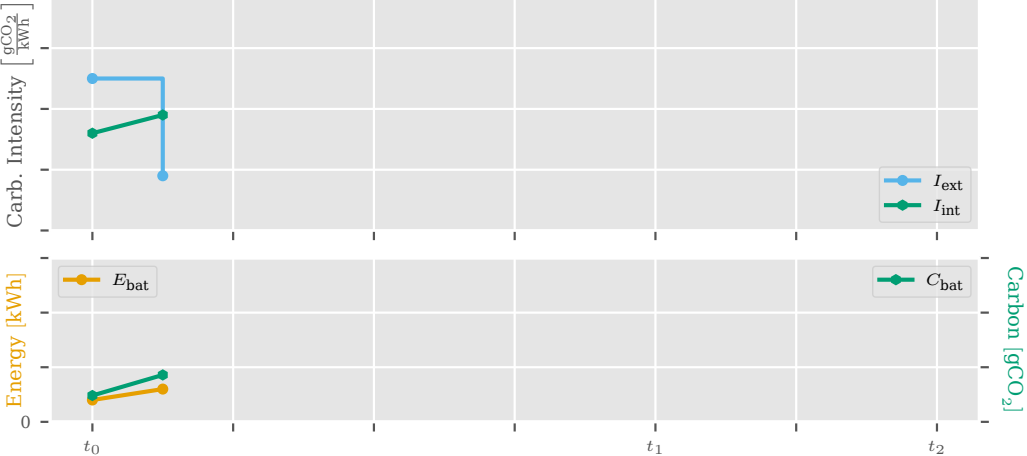
Enabling Carbon Awareness through an Operating-System Daemon (carbond):

- collecting + providing carbon data
- located within the operating system
- currently implementing
 - carbond
 - cperf
 - Energy/Carbon Flame Graphs

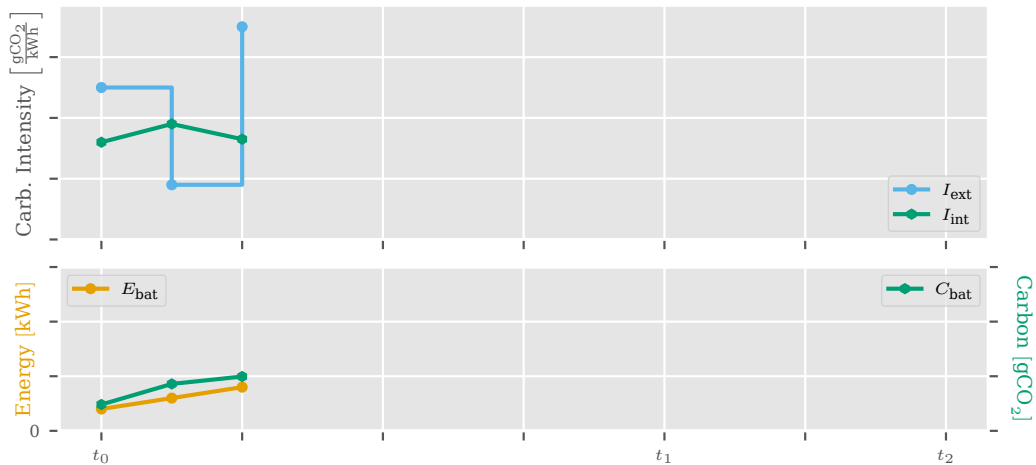
Bochum Operating Systems
and System Software Group



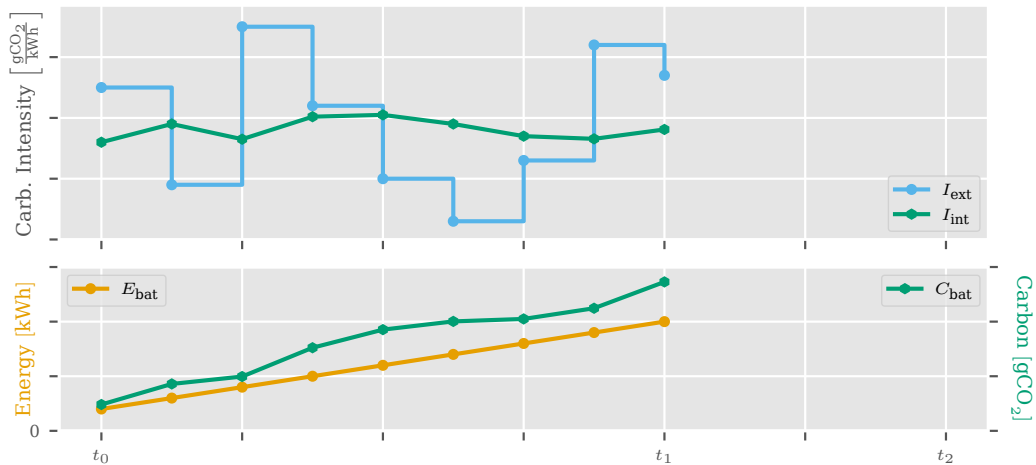
Battery Model



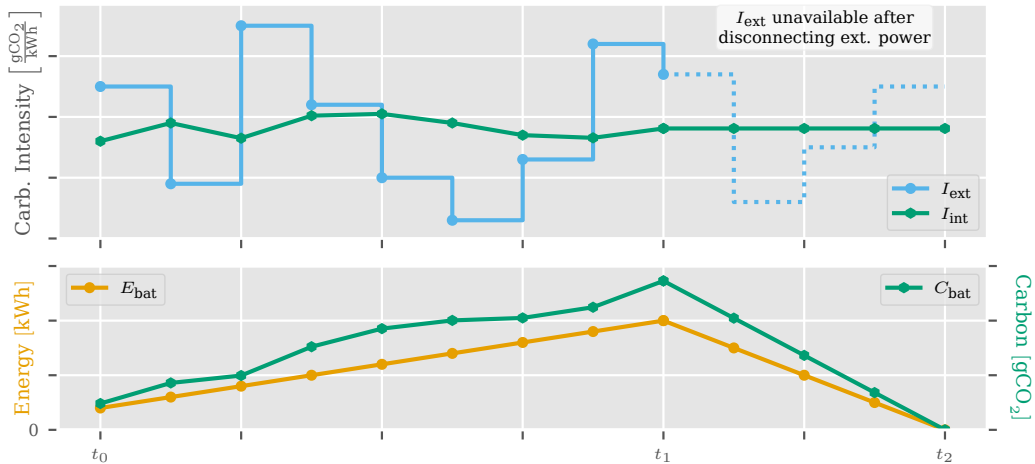
Battery Model




Battery Model



Battery Model



-  Multicians.
645 system: Artist's conception, 2023.
Accessed 03. July 2023.