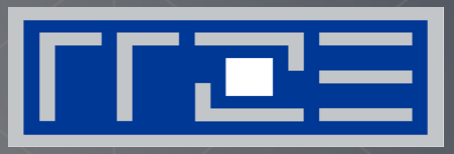


DEMO LIKWID Monitoring Stack (LMS)

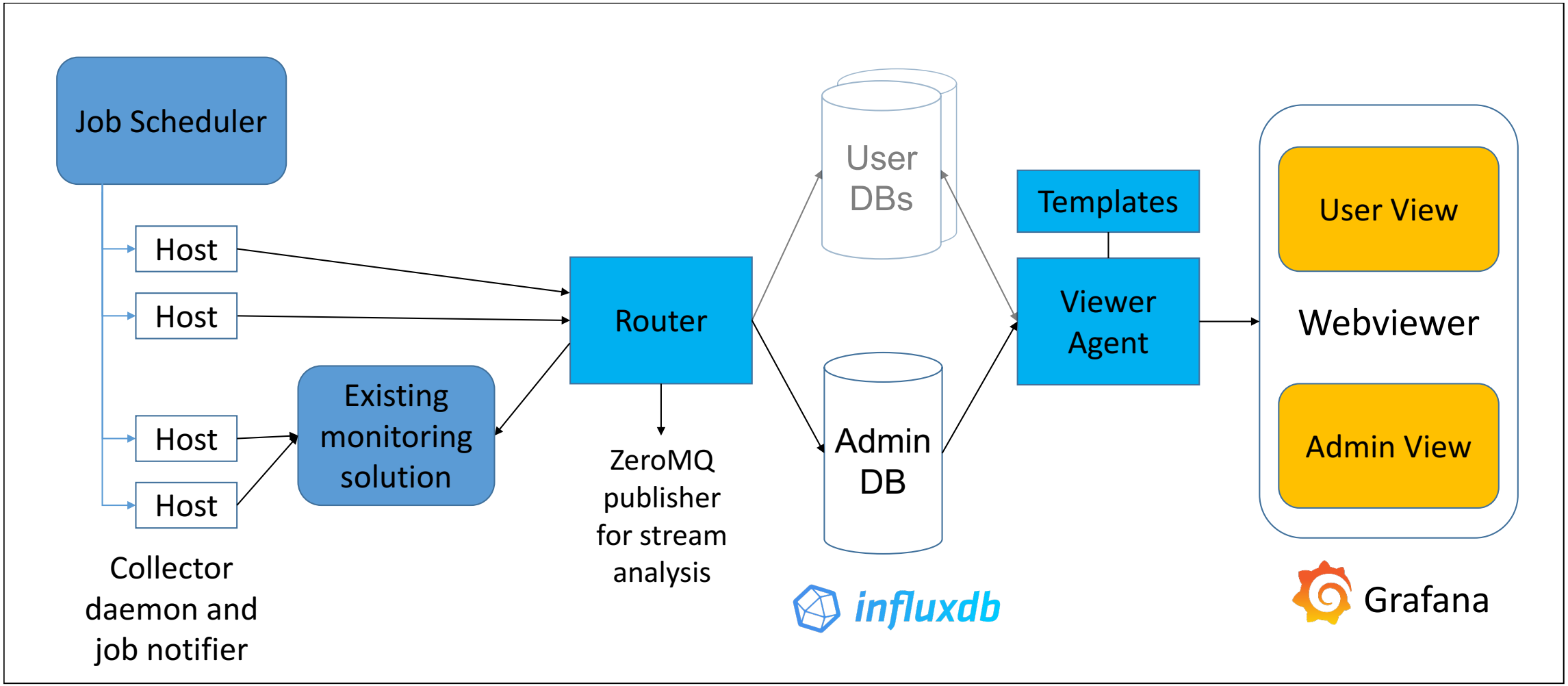
Thomas Röhl

FEPA Workshop, 20.07.2017, RRZE, Erlangen

What‘s the LMS

- Simple and pragmatic
- All layers use default open-source software
- All communication is plain HTTP
- State-of-the-art time-series database(s)
- No integration in job scheduler
- Frontend views are automatically generated
- Caring about data security
- Thought about scalability by allowing hierarchical router architecture

System architecture



DEMO

Job evaluation (Not in DEMO)

- On-The-Fly or after job finishes?
- Current version analyzes job in the end (full access to data)

Job evaluation

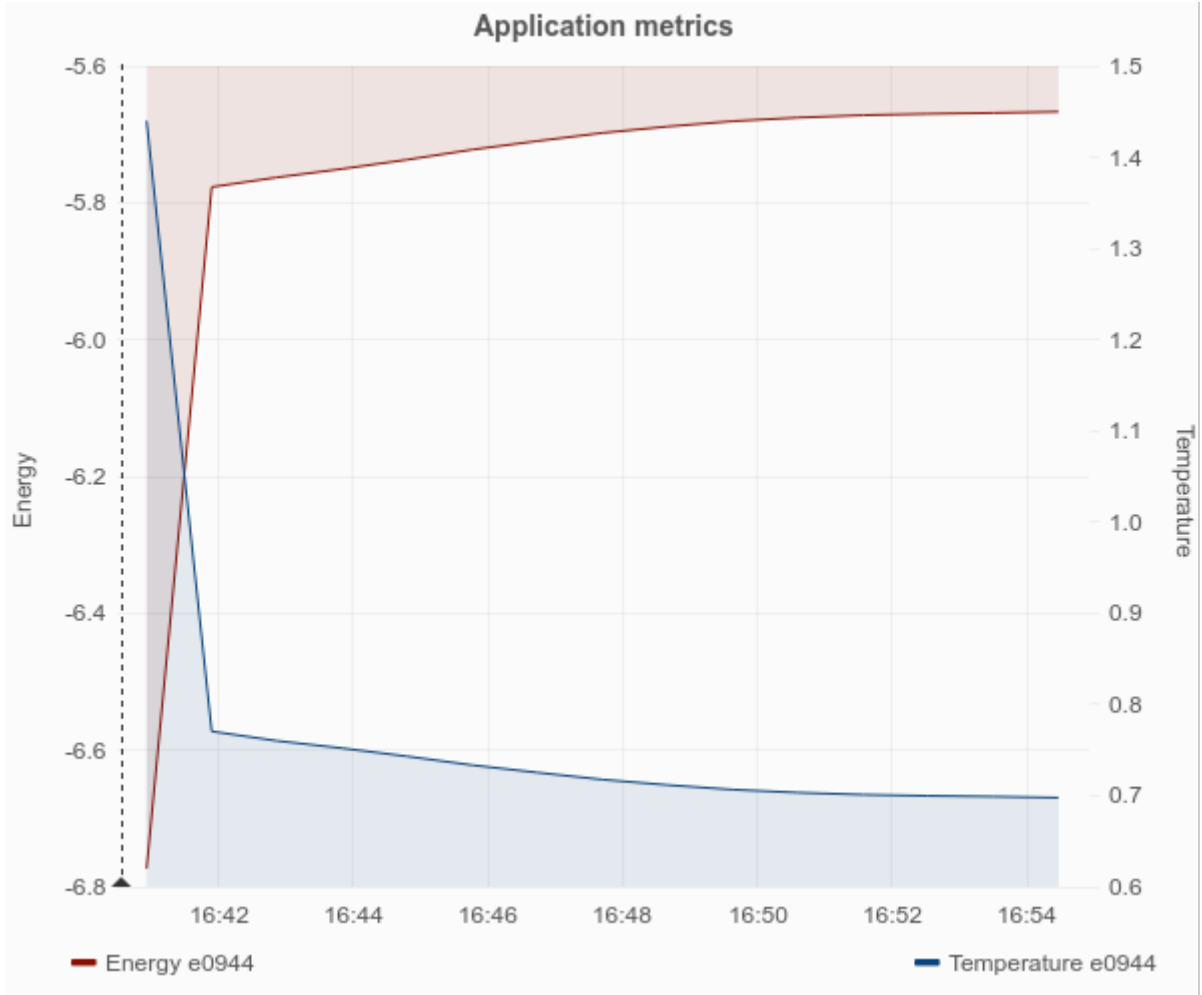
Check	e0814	e0815	e0816	e0817
Memory bandwidth [MByte/s]	low (263.6)	low (620.8)	low (353.9)	low (1552.5)
DP FLOP rate [MFLOP/s]	ok (493.5)	ok (1391.8)	ok (1058.9)	ok (8445.6)
SP FLOP rate [MFLOP/s]	low (0.0)	low (0.0)	low (0.0)	low (0.0)
Lustre FS rate [MByte/s]	none	none	none	none
InfiniBand events [Mevents/s]	good (7.8)	good (14.4)	good (11.7)	good (17.5)
Load Imbalance	bad (Memory bandwidth, DP FLOP rate, InfiniBand events)			

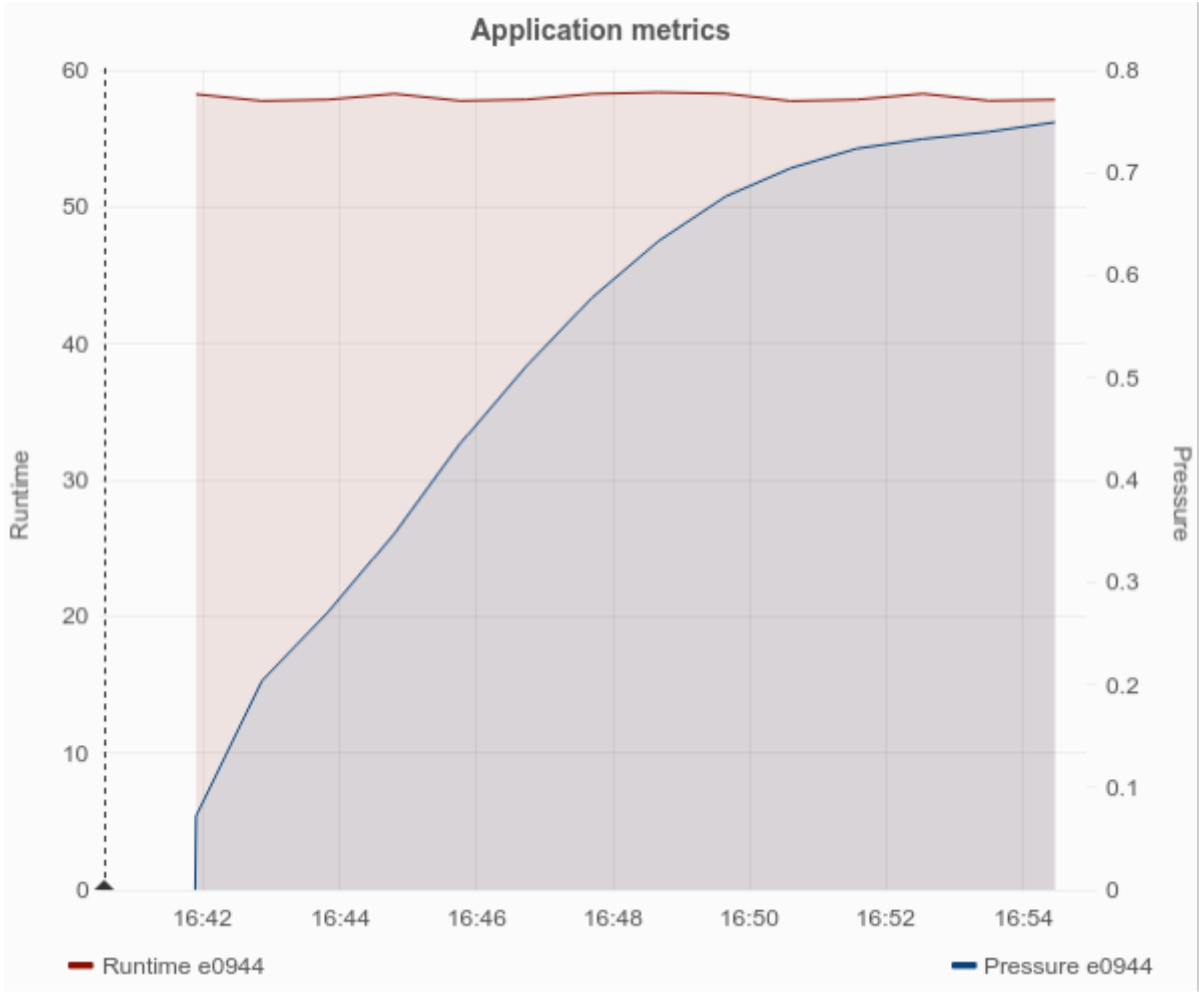
- New version analyzes jobs on the fly (ZeroMQ publisher from router)

Application-level Monitoring (Not in DEMO)

- Small C-Library for instrumenting applications to send
  - Metrics like LUP/s, runtime_per_iteration, …
  - Events like “starting application X“, „entering OpenMP parallel region“
- Command line application exists
- Integration in Shell possible (event for each bash command)
- Script to create overload libraries that send data at function calls:
  - GCC OpenMP
  - Intel/LLVM OpenMP
  - Data-related functions (malloc, free, realloc, …)
  - Affinity related functions (pinning of processes/threads)

Application-level Monitoring (Not in DEMO)





Mantevo miniMD

Danke für die Aufmerksamkeit

LMS: https://github.com/RRZE-HPC/LMS