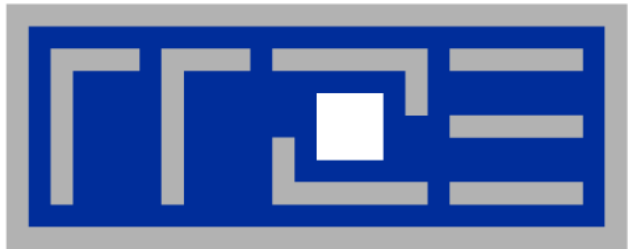


FEPA – A framework for systematic energy and performance analysis of extreme-scale applications in HPC computing centers

T. Röhl
J. Eitzinger
RRZE

a joint project with:

lrz

NEC

Bundesministerium
für Bildung
und Forschung

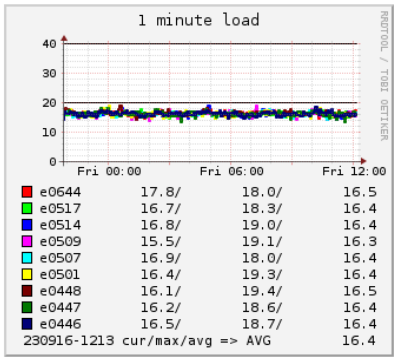
Grant Nr. 01IH13009

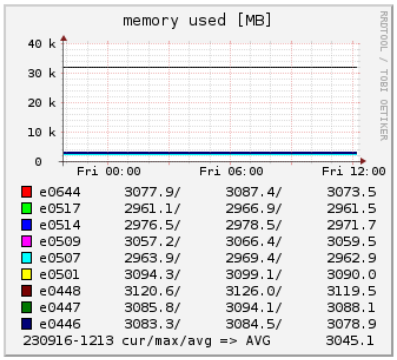
Current/old monitoring system @RRZE

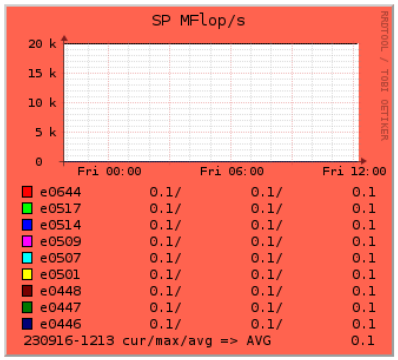
- **Based on Ganglia**
- **Cronjob reads host data periodically**
- **Ganglia stores values in one RRD per metric**
- **Historically grown feature set with distinct pages**
  - Overview about all running jobs and one page per job
  - Host dedicated page without job information
  - Roofline diagram and custom plot generator
  - Maui reservations, NEC InfiniBand wizard
  - GPU overview page for all GPU nodes
- **No (automatic) quality evaluation of job data**
  - only thresholds and human detector

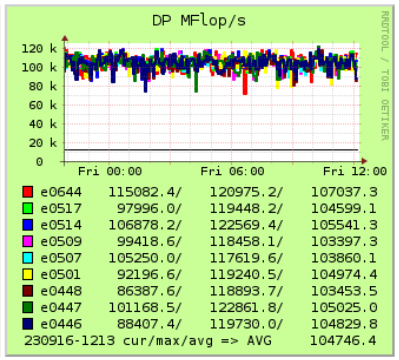
Overview about all running jobs & job-specific page

653066
[redacted]
GIC_massive_1_23
9 nodes
walltime 23:59:00
usedtime 14:27:11
work

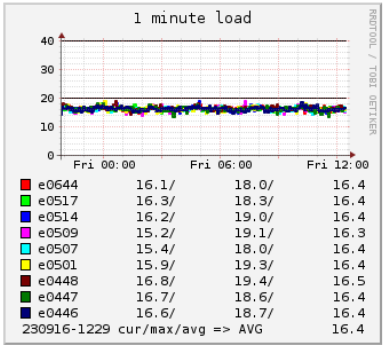


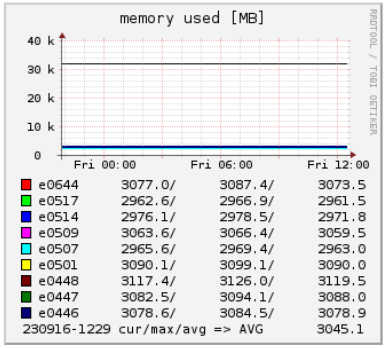


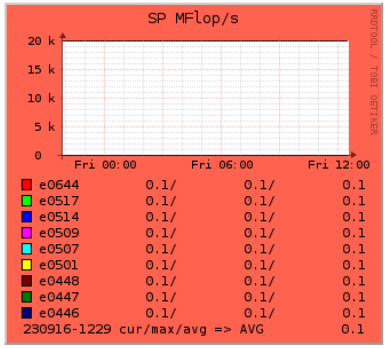


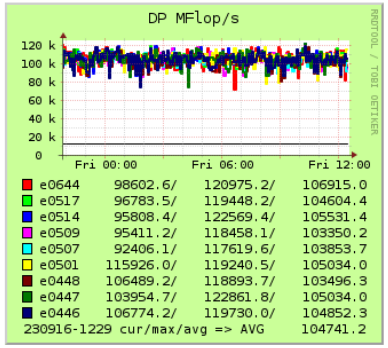


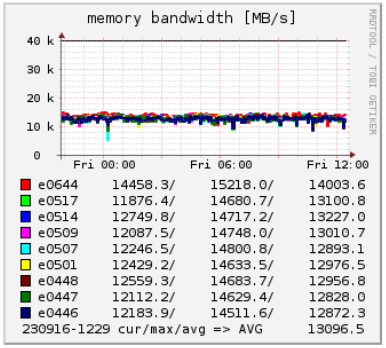
Performance statistic jobid=653066 ([redacted])

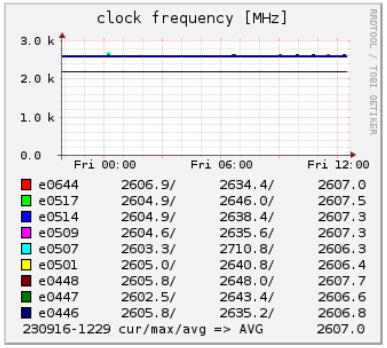


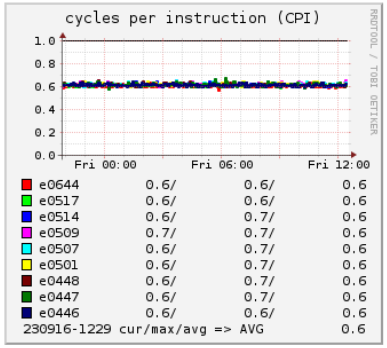


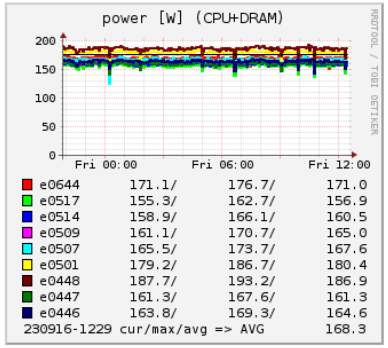




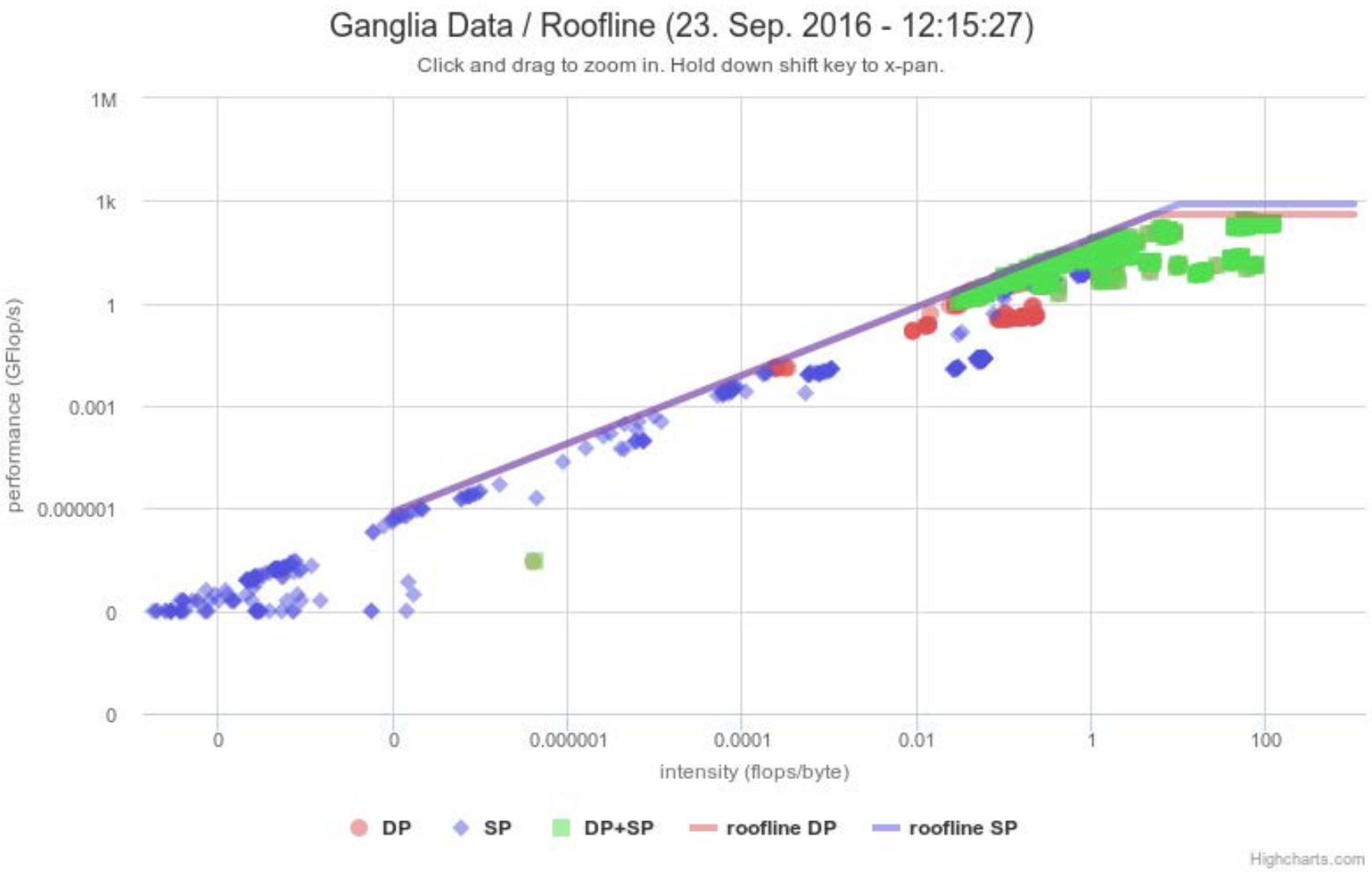




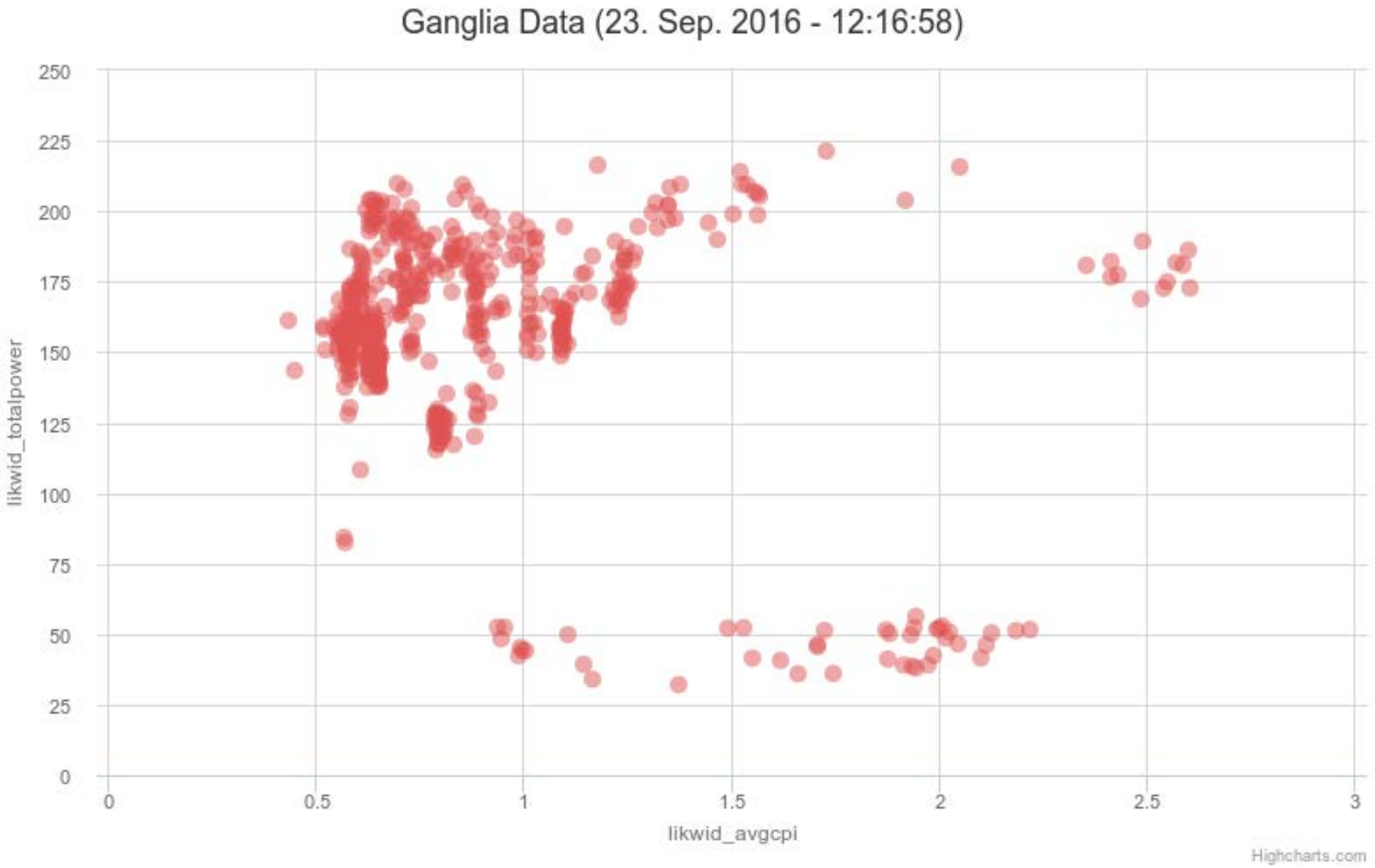




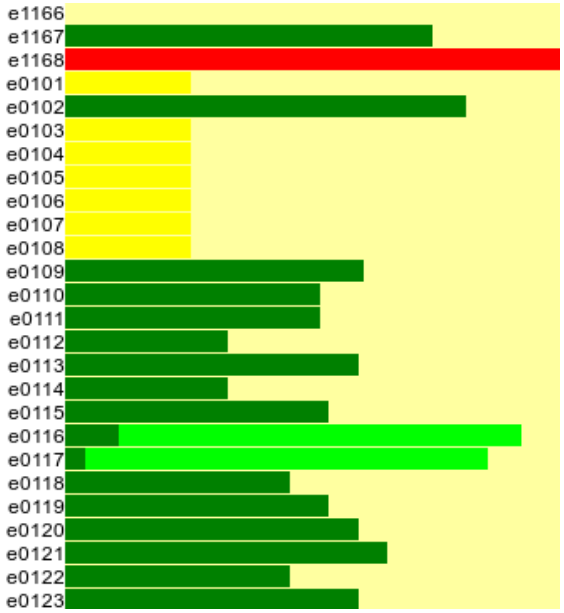
Roofline diagram



Custom plot generator



Maui reservations & Cluster usage status



Node status @ 23.09.2016 - 12:18

1061 _	1062 _	1063 _	1064 _	1065 _	1066 _	1067 _	1068 _	1161 _	1162 _	1163 _	1164 _	1165 _	1166 _	1167 J	1168 X
0101 *	0102 _	0103 *	0104 *	0105 *	0106 *	0107 *	0108 *	0109 J	0110 J	0111 J	0112 J	0113 J	0114 J	0115 J	0116 J
0117 J	0118 J	0119 J	0120 J	0121 J	0122 J	0123 J	0124 J	0125 J	0126 J	0127 J	0128 J	0129 J	0130 J	0131 J	0132 J
0133 J	0134 J	0135 J	0136 J	0137 J	0138 J	0139 J	0140 J	0141 J	0142 J	0143 J	0144 J	0145 J	0146 J	0147 J	0148 J
0149 J	0150 *	0151 J	0152 J	0153 J	0154 J	0155 J	0156 J	0201 J	0202 J	0203 J	0204 J	0205 J	0206 J	0207 J	0208 J
0209 J	0210 J	0211 J	0212 J	0213 J	0214 J	0215 J	0216 J	0217 J	0218 J	0219 J	0220 J	0221 J	0222 J	0223 J	0224 J
0225 J	0226 J	0227 J	0228 J	0229 J	0230 J	0231 J	0232 J	0233 J	0234 J	0235 J	0236 J	0237 J	0238 J	0239 J	0240 J
0241 J	0242 J	0243 J	0244 J	0245 J	0246 J	0247 J	0248 J	0249 J	0250 J	0251 J	0252 J	0301 J	0302 J	0303 J	0304 J

X down	1	. offline	0	J job-exclusive	517	? state-unknown	0
* free-and-error	16	_ free	25	# job-and-error	1	j not-all-procs-used	0

Conclusion for current system(s)

- **Ganglia simple and easy but only admin-view**
  - Job-view must be scripted
- **On HPC systems often vendor-specific tools are used**
  - Newest cluster at RRZE uses Megware‘s ClustWare
  - IMHO: Designed some years ago
- **Just presentation of metrics, no further processing**
- **Less flexible**
- **Reduced feature set compensated with fancy web pages**
- **No spontaneous addition of additional metrics**

Goals of FEPA

- **Provide tooling infrastructure which allows to globally profile application performance in large supercomputing centers**

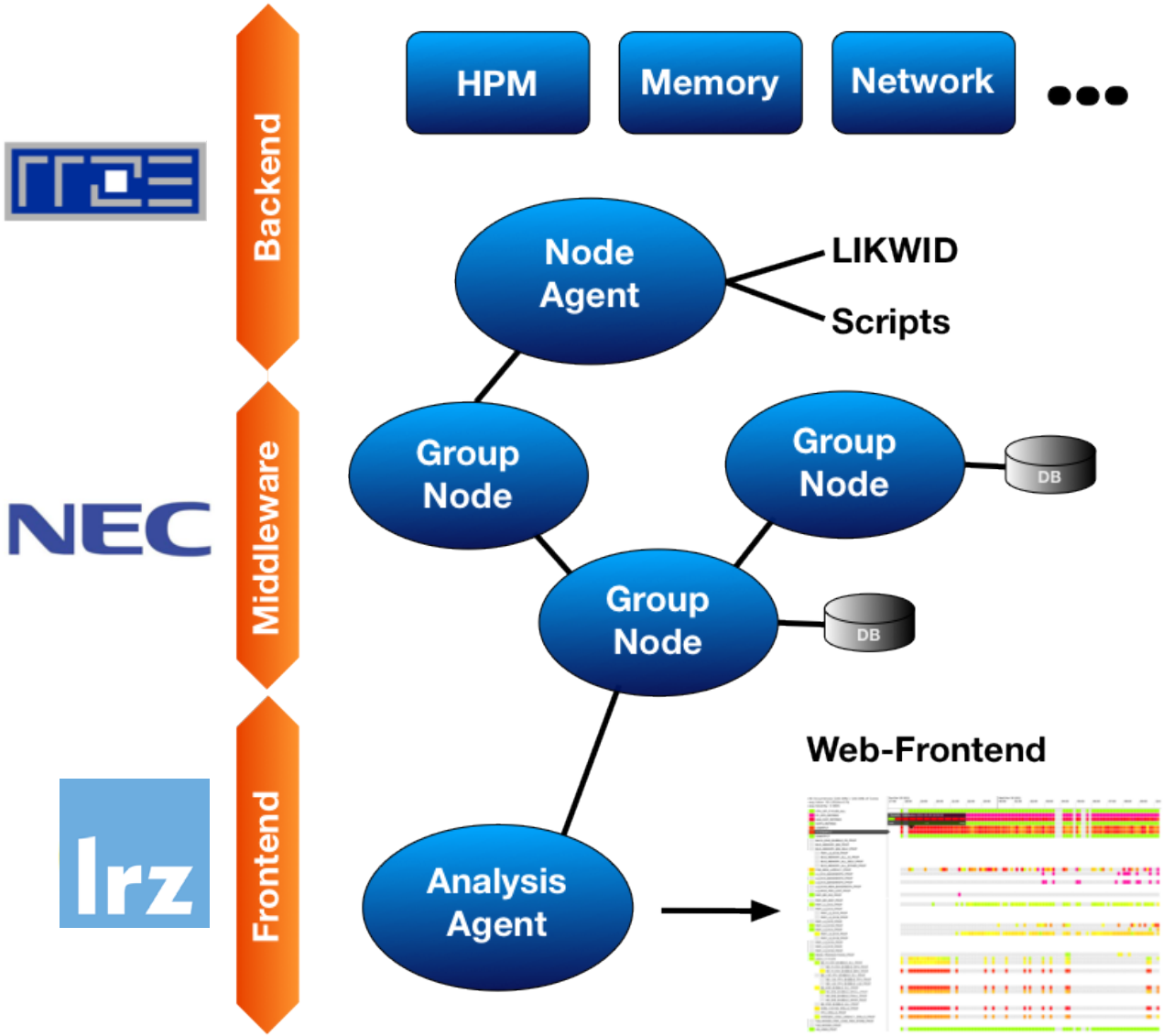
  Scalable, flexible integration of tools, aggregation for multi-device jobs

- **Embed application profiling in a pattern-driven performance engineering process aiming for maximum resource utilization**

  Adaption of metrics required, application level tools provide metrics, hardware performance monitoring, additional hardware information

- **Provide knowledge which enables to significantly improve the efficient use of HPC compute resources across all application domains**

Planned infrastructure



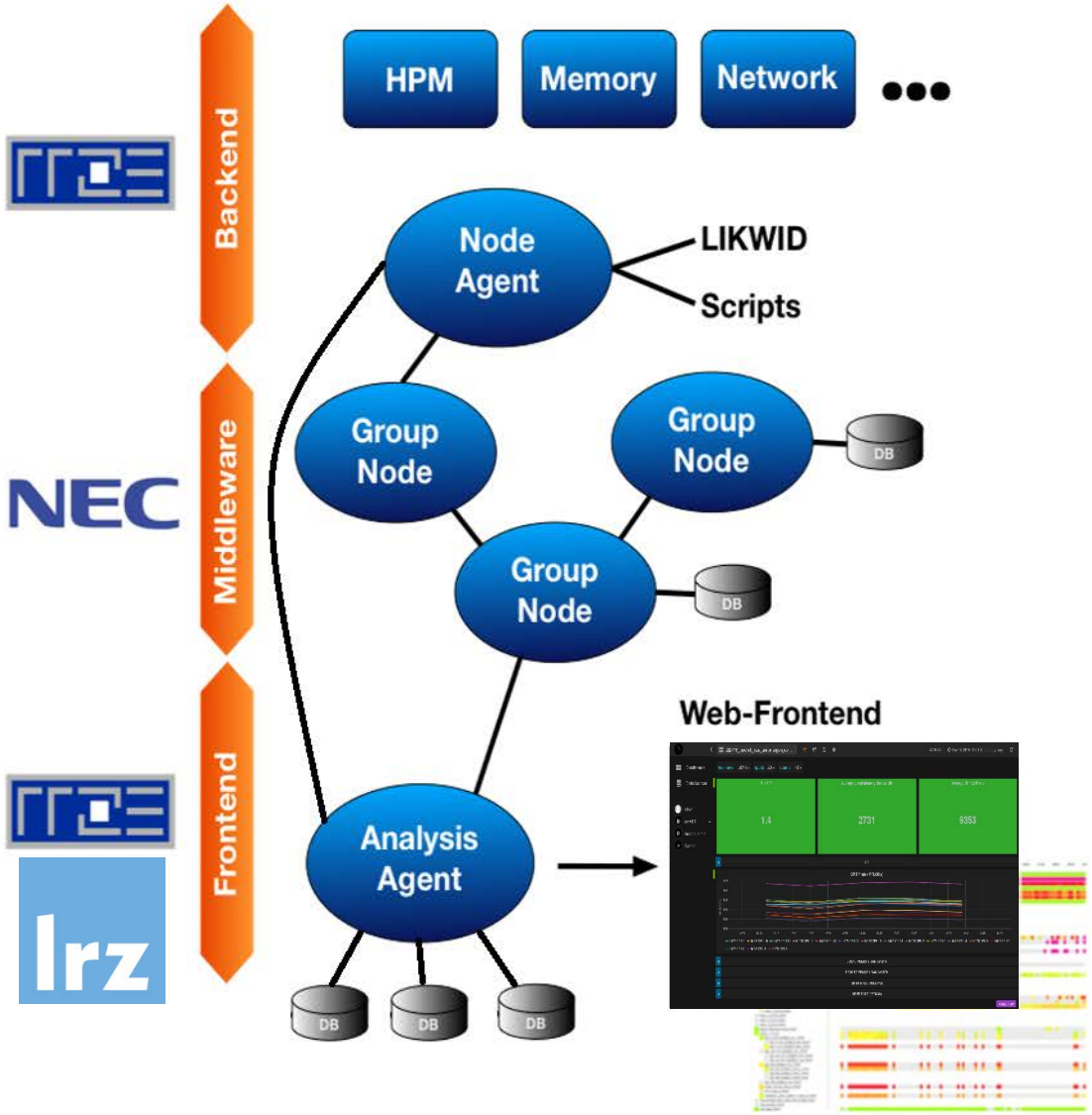
Builds upon the results of previous BMBF projects:

- ISAR (LRZ)
- TIMaCS (NEC)

Opportunity to establish the LIKWID Open Source project as an alternative to established solutions

All components will be Open Source and can also be used stand alone

Realized infrastructure



Basic architecture with aggmon (NEC) middleware for job-wide aggregations

Middleware can by be bypassed for small systems (no aggregations 'on the way')

Each user gets own database + a global admin database (InfluxDB)

Web frontend changed to Grafana (scriptable)

LRZ frontend still in development (required for SuperMUC successor)

Still anything Open-Source

New analysis agent

- **Too much effort to adapt analysis agent of LRZ for RRZE**
- **New analysis agent must perform multiple tasks:**
  - Receive and handle START_JOB and END_JOB signals
    (or get information from job scheduler)
  - Tag all measurements during forward to admin DB
  - Due to security problems duplicate messages in user-job-specific DBs
  - Configure Webfrontend for cluster users
  - Create job-specific Webfrontend views (pre-configured + variable)
    - Traffic lights of selected metrics
    - Job-specific graphs
    - Host-specific graphs
    - CPU-specific graphs

Coarse view

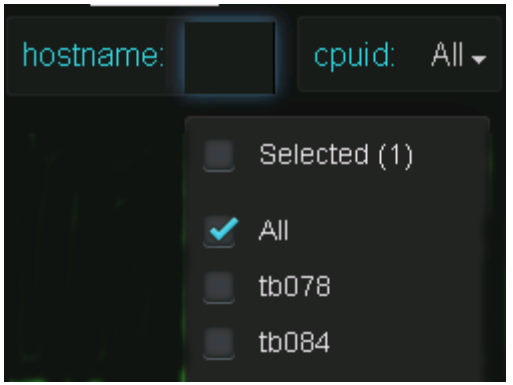
Detailed view

New analysis agent: Further development

- **Provide more options for dashboard creation**
- **Abstract database layer**
  - Support multiple databases
  - Differentiation between time-series and meta data
- **Automatic job evaluation based on more complex thresholds, aggregation and metrics**
- **Interfaces for evaluation (ZeroMQ publishers)**
  - On streaming data (during forward)
  - On full data (at END_JOB)

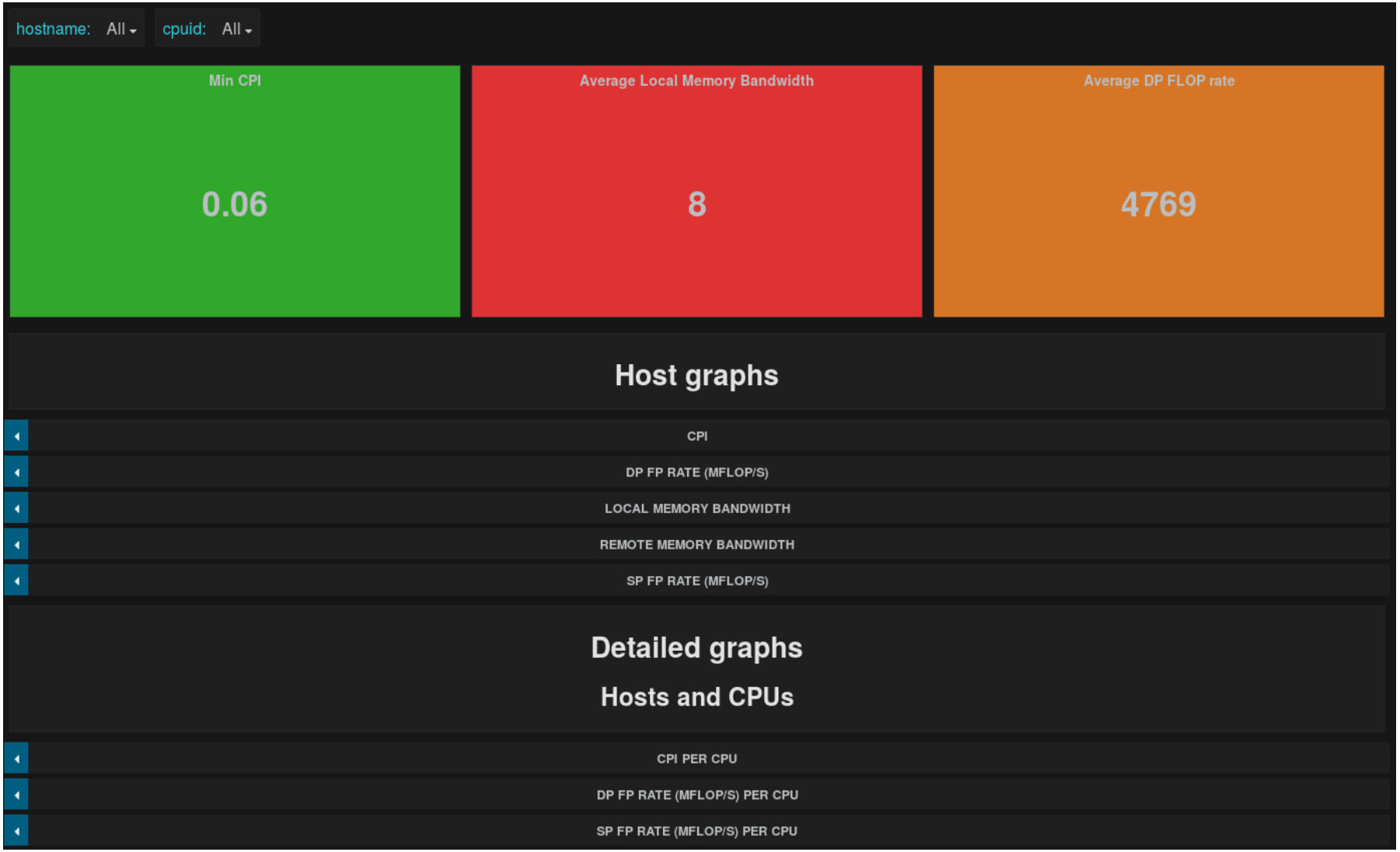
Webfrontend Grafana

- **Commonly visualization of time-series data**
- **Most time-series databases supported**
- **(Almost) anything can be managed through HTTP API**
- **User-management with LDAP or stand-alone**
- **Dashboard configuration with JSON documents**
  - Templates offer easy selection (all hosts of a job, …)
  - Annotations for user/application-defined events



- **But: Main focus on real-time data visualization**
- **To play around: http://play.grafana.org/**

Screenshot (I)



Application-defined metrics (I)

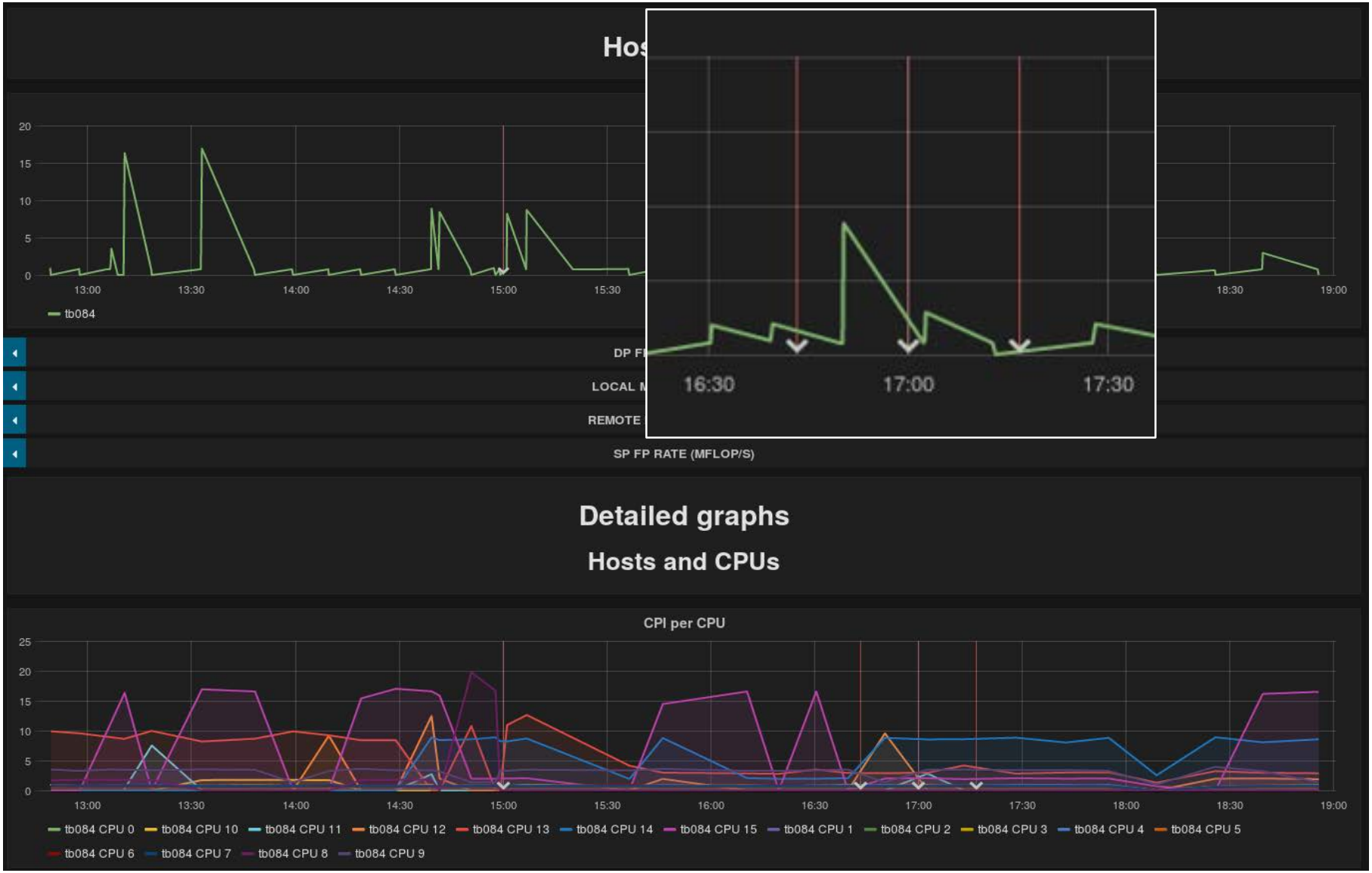
- **Application metrics are essential**
- **Could use big annotation frameworks like Caliper**

  Often: Too much overhead, too big memory footprint and too many features
- **Required:**
  - Simple and pragmatic
  - Minimal memory footprint
  - Flexible to fit into current infrastructure

Application-defined metrics (II)

- **Result libusermetrics:**
  - C library with
    - `init()`
    - `send_values()  // provide metric values`
    - `send_events()  // provide annotations`
    - `close()`
  - Takes arbitrary tags to describe values/event
    - Except ‘user‘, ‘host‘ and ‘jobid‘
  - Restricted tag keys ‘user‘, ‘host‘ and ‘jobid‘ added automatically
  - Shell application exists to supply values/events from command line/job script
  - Values and events are automatically handled by analysis agent
    - Grafana dashboards get extra user metrics section
    - Annotation visible in all graphs
  - Can be embedded in other applications

Screenshot (II)



Current status

- **Installing system on „old“ compute cluster at RRZE**
- **Running installation on some nodes at RWTH Aachen**

- **Integrate pygrafana into analysis agent**
- **Optional connection to job scheduling system**
- **Multiple jobs running on the same node**
  - requires at least pinning
  - How to handle system-wide metrics (memory consumption, load, …)

- **Rethinking whole system with respect to the upcomming DFG project ‚ProPE‘ in collaboration with RWTH Aachen and TU Dresden**
  - Alternative to InfluxDB
  - Split information bases in time-series and meta data
  - Substitue host agent with more flexible solution e.g. Intel Snap

What is not implemented

- **Pattern based performance analysis**

  Reason 1: Measure many HPM events on compute nodes (Overhead)

  Reason 2: Local analysis steals compute power, remote analysis too late to decide which group should be measured next

- **Automatic metric /event submission using MPI_T , OMPT or DWARFs**
- **Performance model graphs for Grafana (Roofline, ECM)**
- **Feedback of analysis agent back to compute nodes:**
  - Adapt CPU frequency (integrate energy-awareness)
  - Change measurement intervals of collectors
  - Change event set for LIKWID

Thank you for your attention!

Any Questions?

Want to try it, contact me

Node Agent: https://github.com/TomTheBear/Diamond

- LIKWID: https://github.com/RRZE-HPC/likwid
- LIKWID Python interface: https://github.com/TomTheBear/likwid-python-api

Middleware: https://github.com/efocht/aggmon (MongoDB/TokuMX)

Analysis agent: https://github.com/TomTheBear/influxdb-grafana-router

Grafana Frontend: https://github.com/grafana/grafana

- Pygrafana: https://github.com/TomTheBear/pygrafana

Persyst Frontend: https://www.webapps.lrz.de/ (with DEMO)

Database: https://github.com/influxdata/influxdb