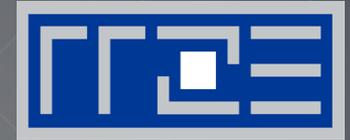


ERLANGEN REGIONAL COMPUTING CENTER



FEPA 

**Ein flexibles Framework zur Energie-
und Performanceanalyse hochparalleler
Applikationen im Rechenzentrum**

J. Eitzinger, T. Röhl, W. Hesse, A. Jeutter, E. Focht
20.7.2017



Motivation

- System Administratoren verwenden Monitoring um
 - Hardwareausfälle und fehlerhaften Betrieb zu erkennen
 - Die Auslastung des Systems zu überwachen
- Anwendungsentwickler nutzen (meist GUI) Werkzeuge um die Performance zu profilieren

Ein flexibles Framework zur Energie- und Performanceanalyse hochparalleler Applikationen im Rechenzentrum

Primäres Ziel

Das primäre Ziel des Projekts ist die Entwicklung einer Infrastruktur zur detaillierten Überwachung der beiden wichtigsten Betriebsparameter großer HPC-Systeme: Applikationsperformance und Energieeffizienz.

Projekt Überblick

- 3rd BMBF Call:
HPC-Software für skalierbare Parallelrechner
- **Laufzeit** 1.7.2013 – 31.6.2016
- **Förderkennzeichen** 01IH13009
- **Umfang** 3 Stellen an drei Standorten



Regionales
Rechenzentrum
Erlangen



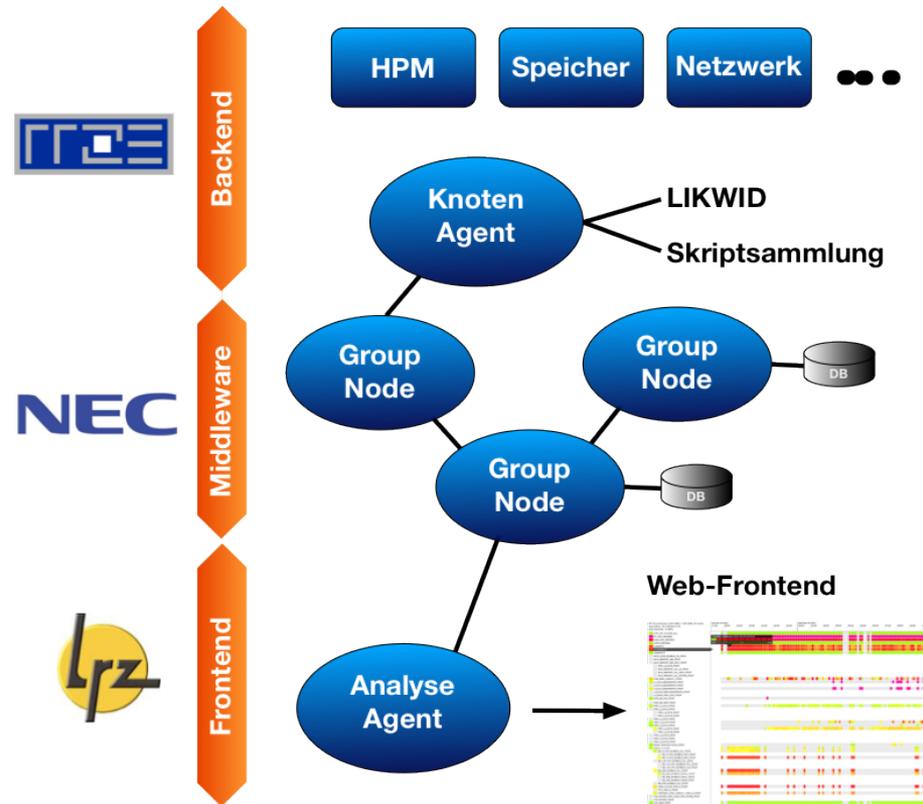
NEC Deutschland
GmbH



Leibniz-
Rechenzentrum

Ziele

- Allow to detect applications with pathological performance behavior
- Help to identify applications with large optimization potential
- Give users feedback about application performance
- Ease access to hardware performance counter data





PROJEKT ERGEBNISSE

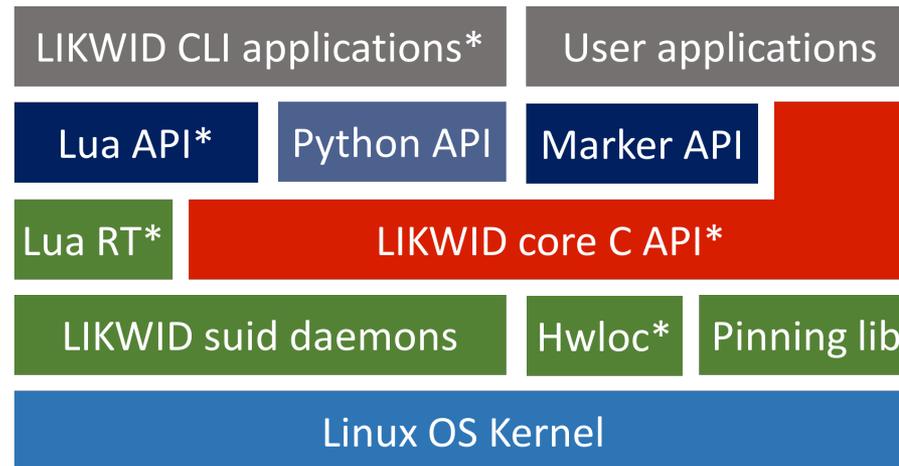


RRZE WP2 Backend

WP 2.1

Umstrukturierung und Erweiterung der LIKWID Bibliothek

- Integration der hwloc Bibliothek zur Erkennung der Knotentopologie
- Skriptschnittstellen für Lua und Python
- Erweiterung aller LIKWID Werkzeuge
- Vollständige Unterstützung der X86 CPUs von Intel: Sandy-Bridge, Ivy-Bridge, Haswell, Broadwell, Skylake, Silvermont, Xeon Phi KNC/KNL
- Portierungen auf IBM Power 8 und ARM Architekturen



*New in LIKWID 4.x

RRZE: Weitere Arbeitspakete

- Integration von LIKWID in Diamond Knotenagenten (WP 2.2)
- Analyse des Overheads bei HPM Messungen (WP 2.3)

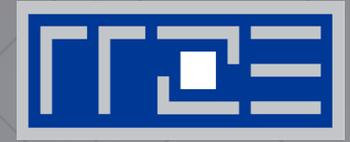
T. Röhl, J. Treibig, G. Hager und G. Wellein, *Overhead Analysis of Performance Counter Measurements*, 43rd International Conference on Parallel Processing Workshops, Sep. 2014, S. 176–185. doi: 10.1109/ICPPW.2014.34.

- Testsuite für Korrektheit der Messungen (WP 2.4)
- Definition und Umsetzung der Performancemuster (WP 2.5)

T. Röhl, J. Eitzinger, G. Hager und G. Wellein, *Validation of Hardware Events for Successful Performance Pattern Identification in High Performance Computing*, Tools for High Performance Computing 2015: Proceedings of the 9th International Workshop on Parallel Tools for High Performance Computing, Dresden, Germany, 2016, S. 17–28. doi: 10.1007/978-3-319-39589-0_2.



PROJEKT ERGEBNISSE



NEC WP3 Middleware

Skalierbare Monitoring Middleware

Erich Focht, Andreas Jeutter

NEC HPC Europe



Motivation: Skalierbarkeit

- Monitoring System soll sowohl auf kleinen als auch auf sehr großen Clustern funktionieren: $O(100-10k)$ Nodes

- verteilt, flexibel, hierarchisch

parallelisierbar, performant aggregierend

- Aggmon

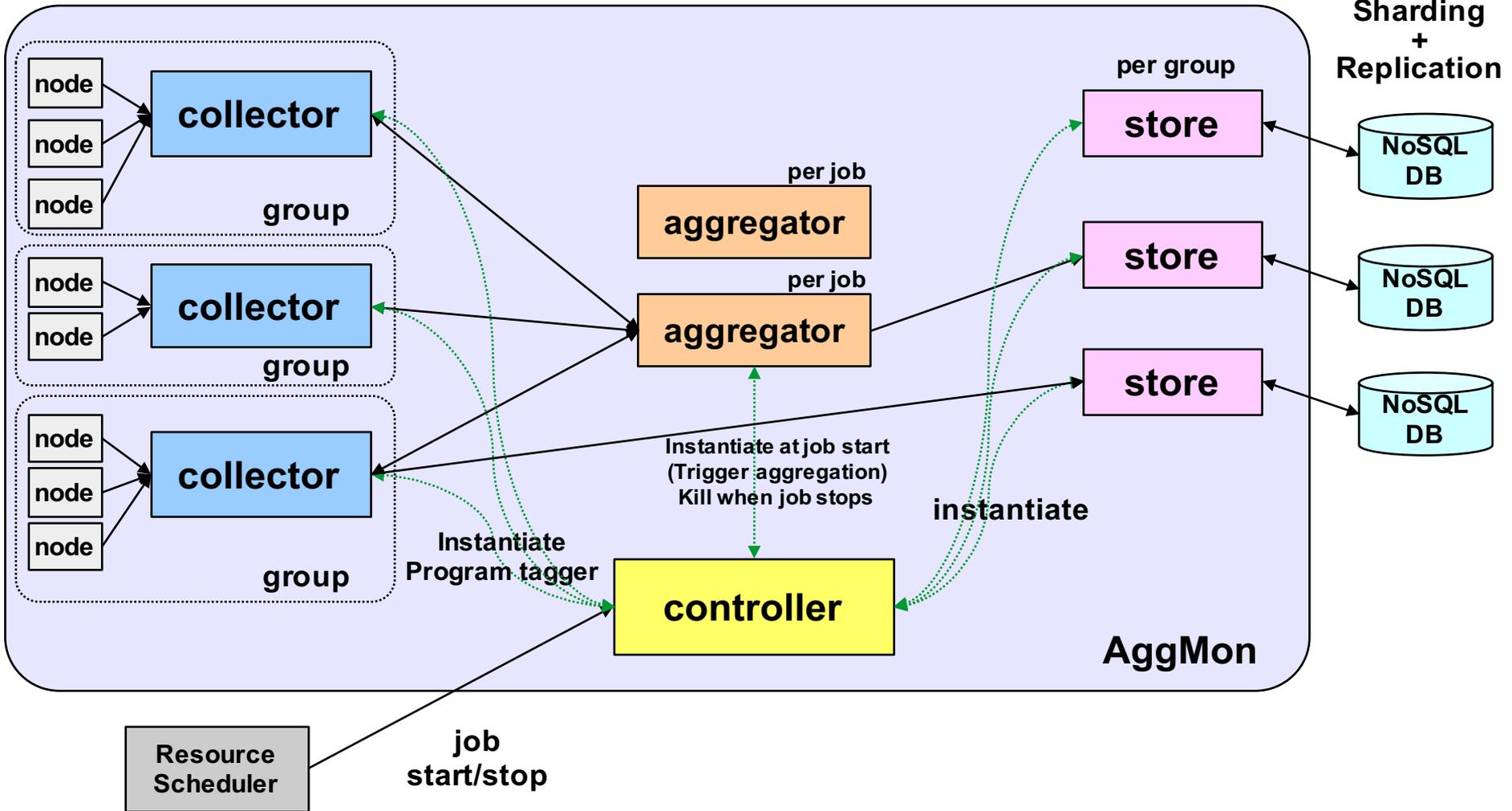
- Im TIMACS BMBF Projekt angefangen (v1)

- Hierarchisch verteilt, skalierbar: Administrationshierarchie der Knoten
- topic basiertes PubSub: AMQP
- Eingebaute flexible Aggregation
- eigene Time Series Datenbank
- Python, monolythisch, GIL verhindert Skalierung

- Im FEPA Projekt weiterentwickelt (v2)

- Administrative Hierarchie
- **Job Hierarchie**

AggMon v2: Architektur



Aggmon v2 in FEPA

- Im FEPA Projekt weiterentwickelt (v2)
 - AMQP durch ZeroMQ ersetzt, Message Rate von O(1k-2k) hoch auf O(50k)
 - Selbst implementiertes PubSub mit flexiblem Matching (PUSH-PULL)
 - RPC Layer auf ZeroMQ: REQ-REP
 - Separate Python Prozesse statt monolythischem Daemon
 - Skalierbar, verteilt, mit ZeroMQ gekoppelt
 - Metrik Datenspeicher: NoSQL Datenbank: MongoDB
 - Eigentlich TokuMX (Percona)
 - Integration von Job Informationen
 - Datenmodell erlaubt Job- und Administrationshierarchie
- Wesentliche Komplikation: Hochverfügbarkeit
 - Viele Ausfallszenarien
 - Controller muss alle Komponenten überwachen und neu starten
 - Fehler in externen Komponenten (Queueing System, Datenbanken)...
 - Controller ist "single point of failure"

Integration

- RRZE
- Quantilaggregation der LIKWID Metriken
 - wie in Persysst am LRZ

TinyBlue Cluster



EMMY Cluster

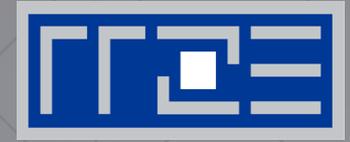


Ausblick: Aggmon v3 jenseits von FEPA

- Ein Controller pro Monitoring Knoten
 - Monitoring Knoten bilden Cluster mit verteiltem Key-Value Store
 - Verteiltes Konsensprotokoll (RAFT)
 - Aufgaben: Komponenten starten, überwachen, neu starten
 - Mehrere Controller pro Hierarchie Knoten teilen sich Aufgaben & Komponenten
- Workflow Routing der Messages mit Hierarchie Tag
- RPC Layer verlegt in den verteilten KV Store
 - Temporär ausgefallene Knoten verpassen Kommandos nicht
- Komponenten Status im KV Store
- Konfiguration & Hierarchien im KV Store
 - Kontrolle des Workflows durch editieren der Hierarchien im KV Store
 - Controller machen den Rest
- Hierarchisches Sharding im Datenbank Backend

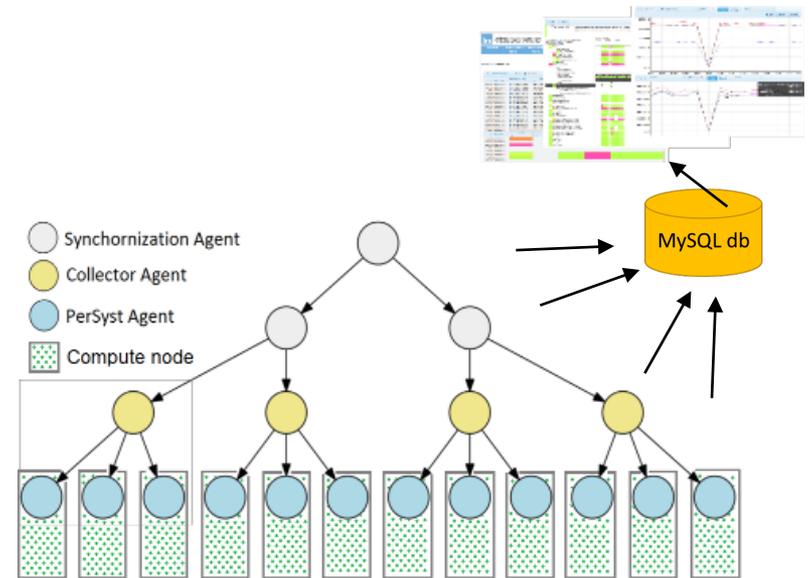


PROJEKT ERGEBNISSE



LRZ WP4 Frontend

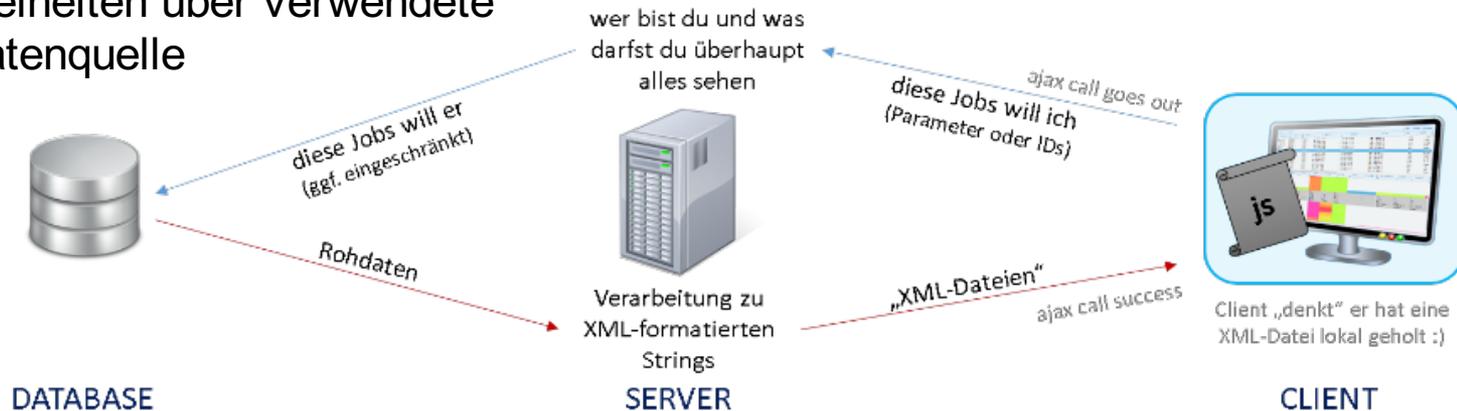
- Verteilte/hierarchische Architektur:
 - PerSyst-Agent (LIKWID-Messung)
 - Collector-Agent (Aggregation)
 - Synchronisations-Agent
- Keine Instrumentierung des Anwender-Codes („läuft nebenbei“)
- Systemweites Monitoring + Analyse
- Korrelation der system-weit gewonnenen Performance/Energie Daten mit den Informationen des Batch Scheduler
- Selektives Monitoring mittels Strategie („Strategy Maps“) ermöglicht Detektion von Bottlenecks und Datenreduktion
- Statt Roh-Daten werden Verteilungen der Performance-Muster (Properties) pro Job gespeichert (Perzentile, Average, Max, Min) -> Datenreduktion
- **MySQL-DB als Schnittstelle zur Visualisierung**





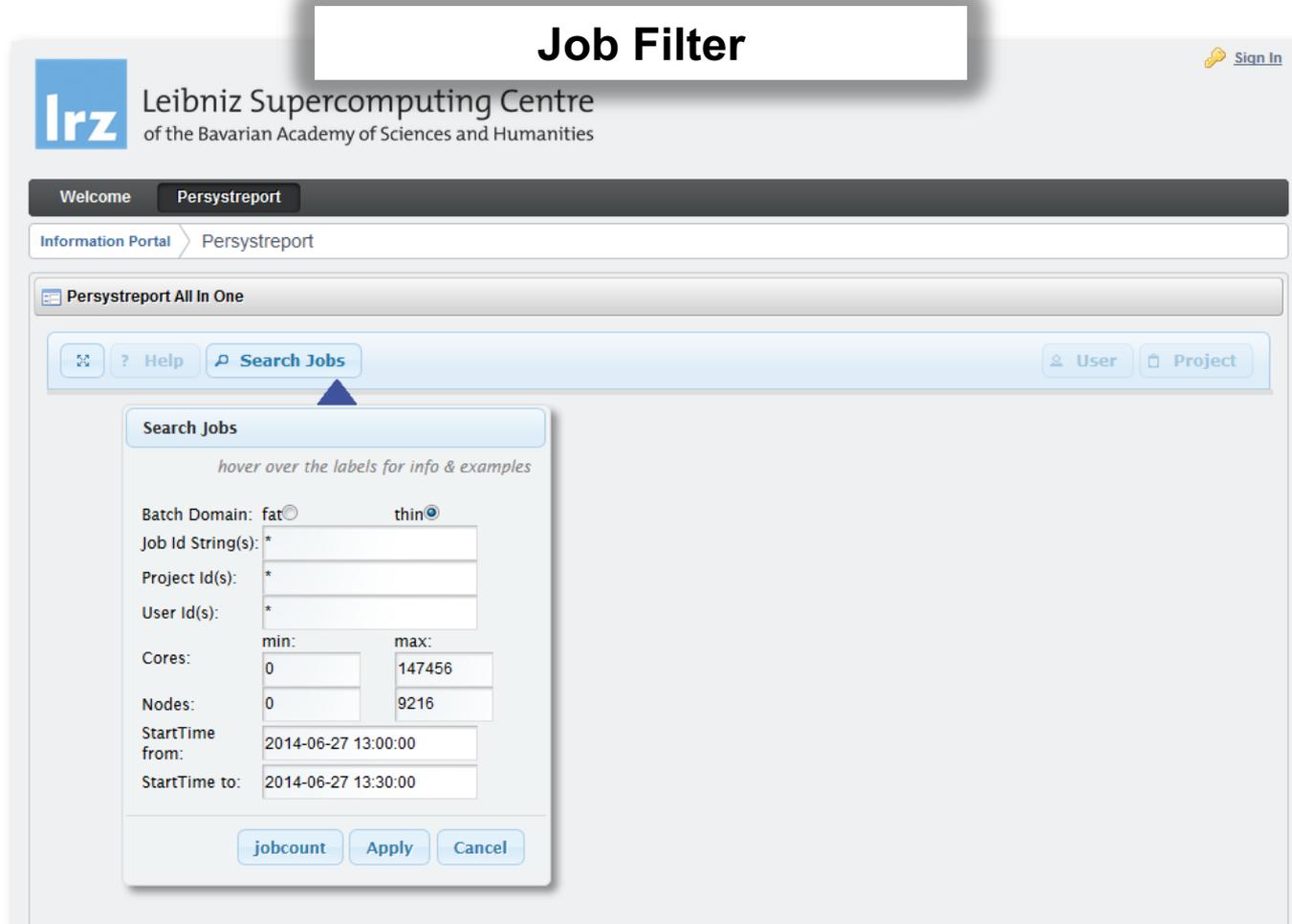
Dynamische Weboberfläche (online):

- Client-Server-System mit Datenbank-Anbindung
 - Der Client = (öffentliche) Internet-Browser und GUI
 - Server = Tomcat-Webserver mit Zugang zu Datenbanken und LDAP-Server
- Schnittstelle zwischen Server und (MySQL-)Datenbank dient JDBC
- Modulare Implementierung: Entkoppelung des Datenbankzugriffcodes vom restlichen Applikationscode
 - Freiheiten über verwendete Datenquelle





Job Filter



Job Filter

Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

Welcome Persysreport

Information Portal > Persysreport

Persysreport All In One

Help Search Jobs User Project

Search Jobs
hover over the labels for info & examples

Batch Domain: fat thin

Job Id String(s): *

Project Id(s): *

User Id(s): *

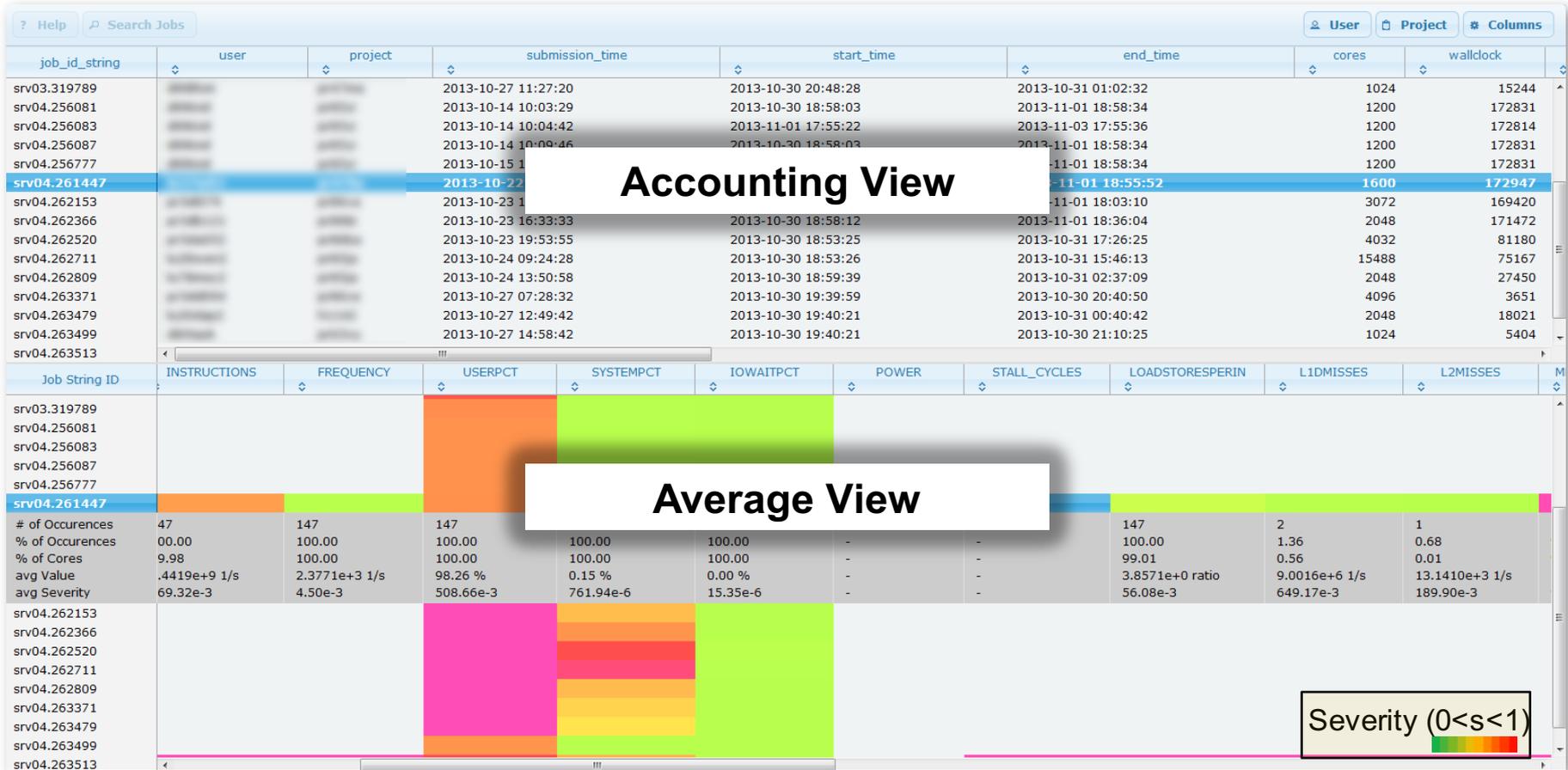
Cores: min: 0 max: 147456

Nodes: 0 9216

StartTime from: 2014-06-27 13:00:00

StartTime to: 2014-06-27 13:30:00

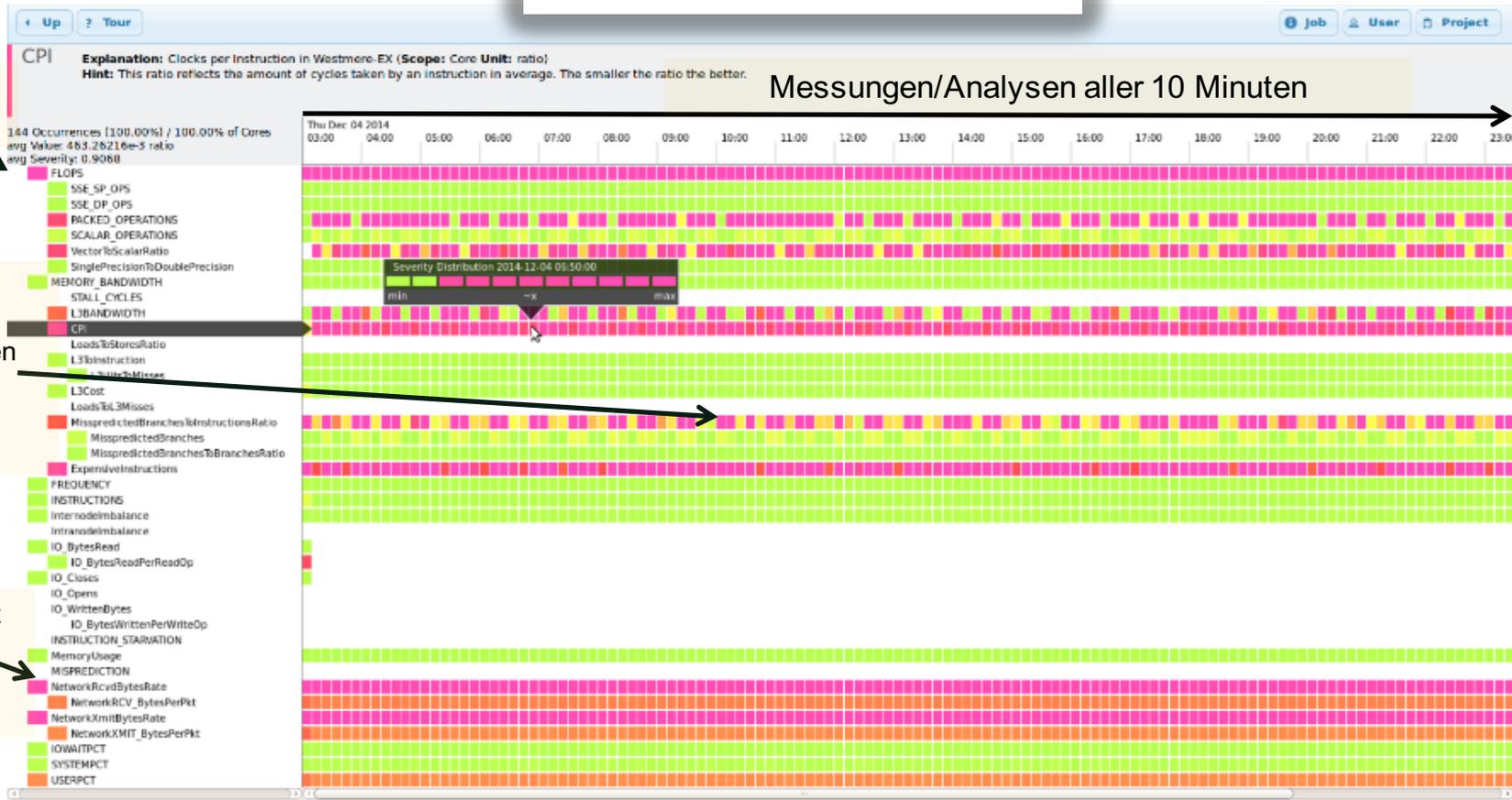
jobcount Apply Cancel





Timeline View

Severity (0<s<1)



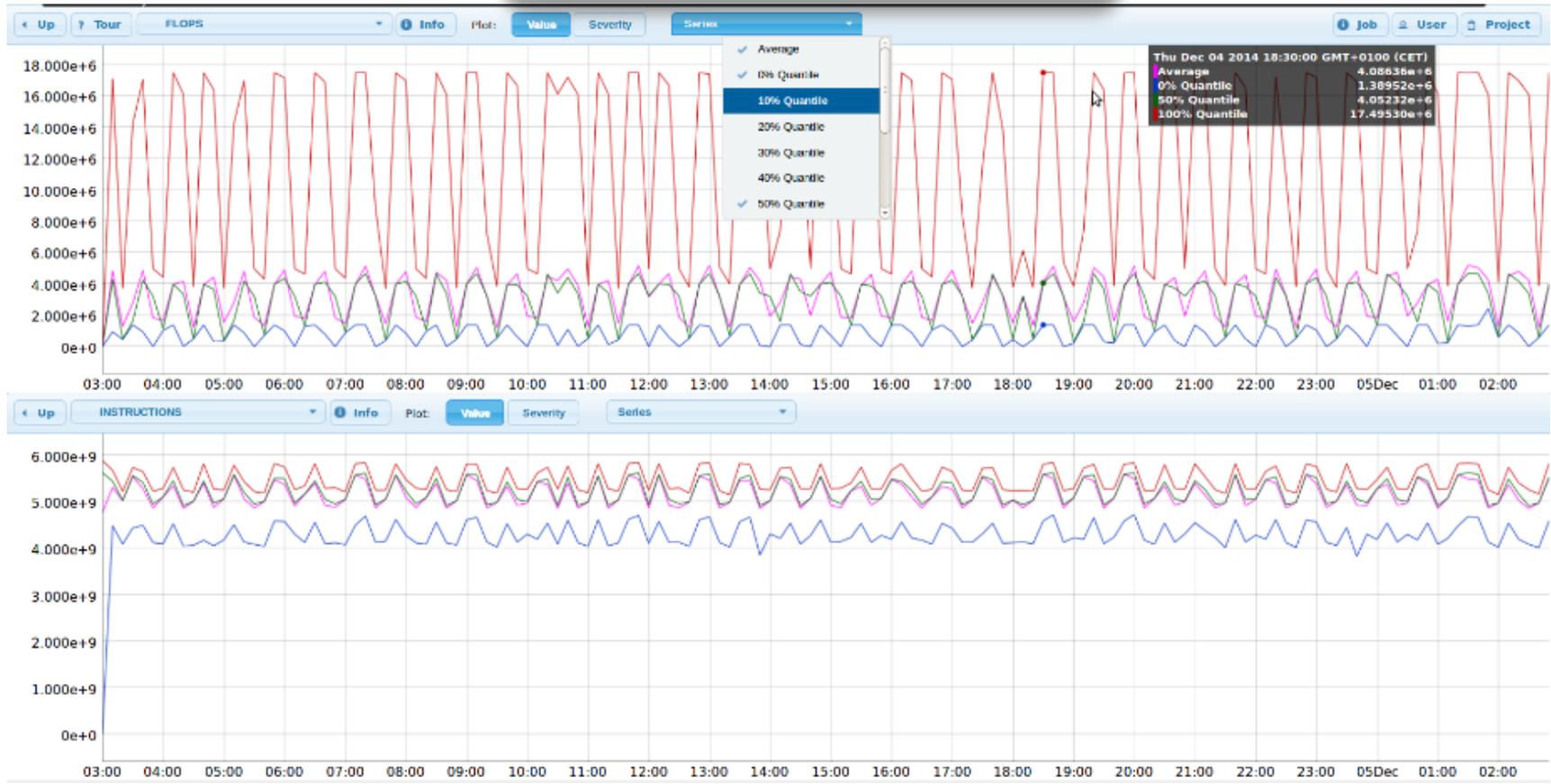
Liste der Properties

Farbe jeder Box entspricht dem Mittelwert über allen Cores bzgl. eines Jobs und einer Messung

Durchschnittswert eines Jobs (bzgl. aller Messungen und Cores)



Performance View



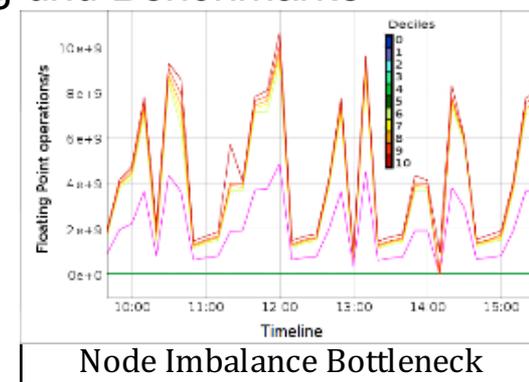
Erfolgreicher Abschluss der Promotion von C. Guillen

Knowledge-based Performance Monitoring for Large Scale HPC Architectures; Dissertation C. Guillen Carias; 2015; <http://mediatum.ub.tum.de?id=1237547>

- **Dokumentation** des PerSyst-Monitoring-System
- **Evaluationsanalysen** zur Bewertung der in PerSyst verwendeten Performancemuster (z.B. für *Flops, Memory Bandwidth, L3 Bandwidth, Load Imbalance Properties, Branches, Branch Misspredictions, I/O Bytes per Operation, Memory Usage*)
- **Statistische Auswertung** der Performancemuster
 - Erläuterungen zur statistischen Berechnung der Performancemuster-Schwellen („thresholds“; notwendig für Strategy Maps)
 - Bem.: Regelmäßige Überprüfung dieser Schwellen während SuperMUC-Produktion

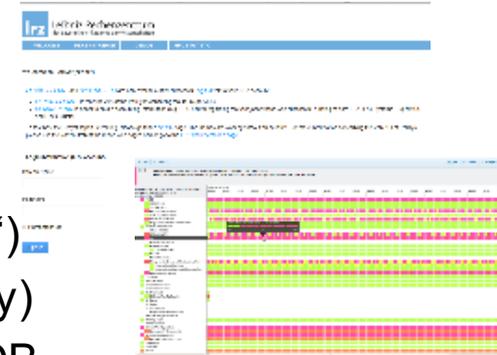
PerSyst-Monitoring ist **produktiv** @SuperMUC Phase I/II

- Definition und Umsetzung der Performancemuster Phase 1 (Westmere-EX, SandyBridge-EP) und Phase 2 (Haswell)
- „Strategy Maps“ (= „Selektives Monitoring durch Verwendung von Strategien“) für beide SuperMUC-Phasen
- Nutzung und Verifikation durch:
 - LRZ-Applikationsunterstützungsgruppe und IBM-Mitarbeiter
 - Benachrichtigung der Benutzer, falls offensichtliche Bottlenecks vorliegen + Vorschläge für Optimierungen
 - Sichtung von Anwendungen für Extreme Scaling und Benchmarks
 - SuperMUC-Benutzer
 - Pos. Feedback bzgl. Nützlichkeit
- Verwendete Strategien (im Wesentlichen):
 - Memory bandwidth analysis,
 - I/O analysis
 - Intra-node/inter-node load imbalance analysis



PerSyst: Status (Teil 3)

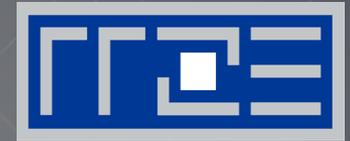
- Visualisierung/Web-Frontend verfügbar für SuperMUC Phase I + II
 - <https://webapps.lrz.de/>
 - Zugang nur für SuperMUC-Benutzer
 - Demos für alle (!) Webseitenbesucher (mit „Tour“)
 - Überarbeitete Sicherheitsrealisierung (Web Proxy)
 - Severity-Berechnung „on the fly“ nicht mehr via DB
 - Flexibilität bei der Portierung des Webfrontendes
 - Datenreduktion (50%) bei gleicher Performance
- Umsetzung des Web-Frontend Softwarepaketes an RRZE Testcluster
 - Krankheitsbedingt nicht abgeschlossen
 - Überarbeitete Software („Build System“, d.h. modulare Programmierung -> RRZE-Modul + Allg. Softwaremodule
 - Anbindung an MongoDB mit Accounting und Likwid-Messdaten



Zusammenfassung und Ausblick

- Alle drei Teilaspekte des FEPA Stacks haben Ihre Ziele erreicht
 - Das Ziel einen integrierten Stack bereitzustellen ist innerhalb der Projektlaufzeit nicht gelungen
 - Verwertung der FEPA Ergebnisse ist jedoch bei allen Projektpartnern gesichert
-
- LRZ PerSyst System ist im produktiven Einsatz
 - Im Rahmen des DFG ProPE Projekts wird am RRZE aufbauend auf den FEPA Ergebnissen der Likwid Monitoring Stack weiterentwickelt <https://github.com/RRZE-HPC/LMS>
 - NEC wird weiter an Ihrem kommerziellen Monitoringsystem arbeiten

ERLANGEN REGIONAL COMPUTING CENTER



Regionales
Rechenzentrum
Erlangen



NEC Deutschland
GmbH



Leibniz-
Rechenzentrum

Thank You.

