PageBeat - Zeitreihenanalyse und Datenbanken

Andreas Finger
Institut für Informatik
Universität Rostock
18051 Rostock
andreas.finger@uni-rostock.de

Ilvio Bruder
Institut für Informatik
Universität Rostock
18051 Rostock
ilvio.bruder@uni-rostock.de

Andreas Heuer
Institut für Informatik
Universität Rostock
18051 Rostock
andreas.heuer@uni-rostock.de

Steffen Konerow
Mandarin Medien GmbH
Graf-Schack-Allee 9
19053 Schwerin
sk@mandarin-medien.de

Martin Klemkow
Mandarin Medien GmbH
Graf-Schack-Allee 9
19053 Schwerin
mk@mandarin-medien.de

ABSTRACT

Zeitreihendaten und deren Analyse sind in vielen Anwendungsbereichen eine wichtiges Mittel zur Bewertung, Steuerung und Vorhersage. Für die Zeitreihenanalyse gibt es eine Vielzahl von Methoden und Techniken, die in Statistiksoftware umgesetzt und heutzutage komfortabel auch ohne eigenen Implementierungsaufwand einsetzbar sind. In den meisten Fällen hat man es mit massenhaft Daten oder auch Datenströmen zu tun. Entsprechend gibt es spezialisierte Management-Tools, wie Data Stream Management Systems für die Verarbeitung von Datenströmen oder Time Series Databases zur Speicherung und Anfrage von Zeitreihen. Der folgende Artikel soll hier zu einen kleinen Überblick geben und insbesondere die Anwendbarkeit an einem Projekt zur Analyse und Vorhersage von Zuständen von Webservern veranschaulichen. Die Herausforderung innerhalb dieses Projekts „PageBeat“ ist es massenhaft Zeitreihen in Echtzeit zu analysieren und für weiterführende Analyseprozesse zu speichern. Außerdem sollen die Ergebnisse zielgruppenspezifisch aufbereitet und visualisiert sowie Benachrichtigungen ausgelöst werden. Der Artikel beschreibt den im Projekt gewählten Ansatz und die dafür eingesetzten Techniken und Werkzeuge.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Big Data, Data Mining and Knowledge Discovery, Streaming Data

In: G. Specht, H. Gamper, F. Klan (eds.): Proceedings of the 26^{th} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 21.10.2014 - 24.10.2014, Bozen, Italy, published at http://ceur-ws.org.

Keywords

Datenanalyse, R, Time Series Database

1. EINFÜHRUNG

Zeitreihen sind natürlich geordnete Folgen von Beobachtungswerten. Die Zeitreihenanalyse beschäftigt sich mit Methoden zur Beschreibung dieser Daten etwa mit dem Ziel der Analyse (Verstehen), Vorhersage oder Kontrolle (Steuerung) der Daten. Entsprechende Methoden stehen in freier und kommerzieller Statistiksoftware wie R[1] , Matlab[2], Weka[3] [7], SPSS[4] und anderen zur Verfügung wodurch eine komfortable Datenauswertung ohne eigenen Implementierungsaufwand ermöglicht wird. Verfahren zur Zeitreihenanalyse sind etwa die Ermittlung von Trends und Saisonalität, wobei der Trend den längerfristigen Anstieg und die Saisonalität wiederkehrende Muster (jedes Jahr zu Weihnachten steigen die Verkäufe) repräsentieren. So werden Abhängigkeiten in den Daten untersucht, welche eine Prognose zukünftiger Werte mit Hilfe geeigneter Modelle ermöglichen. In einer Anwendung die in hoher zeitlicher Auflösung eine Vielzahl von Messwerten erfasst, entstehen schnell große Datenmengen. Diese sollen in Echtzeit analysiert werden und gegebenenfalls zur weiteren Auswertung persistent gespeichert werden. Hierfür existieren zum Einen Ansätze aus der Stromdatenverarbeitung und zum Anderen zur Speicherung von auf Zeitreihen spezialisierte Datenbanksysteme (Time Series Databases). Da statistische Analysen etwa mit standalone R Anwendungen nur funktionieren, solange die zu analysierenden Daten die Größe des Hauptspeichers nicht überschreiten, ist es notwendig die statistische Analyse in Daten-

[1]R – Programmiersprache für statistische Rechnen und Visualisieren von der R Foundation for Statistical Computing, `http://www.r-project.org`.

[2]Matlab – kommerzielle Software zum Lösen Veranschaulichen mathematischer Probleme vom Entwickler The Mathworks, `http://www.mathworks.de`.

[3]Weka – Waikato Environment for Knowledge Analysis, ein Werkzeugkasten für Data Mining und Maschinelles Lernen von der University of Waikato, `http://www.cs.waikato.ac.nz/ml/weka/`.

[4]SPSS – kommerzielle Statistik- und Analysesoftware von IBM, `http://www-01.ibm.com/software/de/analytics/spss`.

banksysteme zu integrieren. Ziel ist dabei der transparente Zugriff auf partitionierte Daten und deren Analyse mittels partitionierter statistischen Modelle. In [6] werden verschiedene Möglichkeiten der Integration beschrieben und sind in Prototypen basierend auf postgreSQL bereits umgesetzt. Auch kommerzielle Produkte wie etwa Oracle R Enterprise[4] integrieren statistische Analyse auf Datenbankebene. Im Open-Source-Bereich existiert eine Vielzahl von Ansätzen zum Umgang mit Zeitreihen, wobei uns InfluxDB[5] als besonders geeignetes Werkzeug aufgefallen ist.
Die Herausforderung innerhalb des im Weiteren beschriebenen Projekts „PageBeat“ ist es innovative und anwendungsreife Open-Source-Lösungen aus den genannten Bereichen zur Verarbeitung großer Zeitreihendaten innerhalb des Projektes miteinander zu kombinieren. Im Folgenden wird das Projekt vorgestellt, um dann verschiedene in Frage kommenden Techniken und abschließend das gewählte Konzept und erste Ergebnisse vorzustellen.

2. PROJEKT PAGEBEAT

Mit „PageBeat“ wird eine als „Software as a Service“ (SAAS) angebotene Softwaresuite speziell zur Beobachtung und Überprüfung von Webanwendungen entwickelt. Dies erfolgt zunächst im Rahmen eines vom Bundeswirtschaftsministerium geförderten ZIM-Kooperationsprojektes. Ziel der Software ist das Beobachten des und das Berichten über den aktuellen technischen Status einer Webanwendung (Website, Content Management System, E-Commerce System, Webservice) sowie das Prognostizieren technischer Probleme anhand geeigneter Indikatoren (Hardware- und Softwarespezifische Parameter). Die Berichte werden dabei für unterschiedliche Nutzergruppen (Systemadministratoren, Softwareentwickler, Abteilungsleiter, Geschäftsführung, Marketing) und deren Anforderungen aufbereitet und präsentiert. Mittels „PageBeat“ werden somit automatisiert Fehlerberichte erstellt, die über akute sowie vorhersehbare kritische Änderungen der Betriebsparameter einer Webanwendung informieren und zielgruppenspezifisch dargestellt werden.
Bei den zugrunde liegenden Kennzahlen handelt es sich um eine Reihe von Daten, die den Zustand des Gesamtsystems im Anwendungsbereich Webshopsysteme widerspiegeln. Dies sind Kennzahlen des Serverbetriebssystems (etwa CPU oder RAM Auslastung) als auch anwendungsspezifische Kenndaten (etwa die Laufzeit von Datenbankanfragen). Diese Daten sind semantisch beschrieben und entsprechende Metadaten sind in einer Wissensbasis abgelegt. Darüber hinaus ist die Verwendung weiterer Kontextinformationen angedacht, die Einfluss auf den technischen Status des Systems haben können. Hierbei kann es sich etwa um Wetterdaten handeln: beim Kinobetreiber Cinestar ist ein regnerisches Wochenende vorausgesagt, dass auf eine hohe Auslastung des Kinokartenonlineshops schließen lässt. Ein anderes Beispiel wären Informationen aus der Softwareentwicklung: bei Codeänderungen mit einem bestimmten Zeitstempel können Effekte in den Auswertungen zu diesem Zeitpunkt nachgewiesen werden. Das Ändern oder Hinzufügen bzw. Beachten von relevanten Inhalten auf den Webseiten können signifikante Änderungen in Analysen ergeben, z.B. bei der Schaltung von Werbung oder bei Filmbewertungen zu neu anlaufenden Filmen auf sozialen Plattformen.

[5]InfluxDB - An open-source distributed time series database with no external dependencies. `http://influxdb.com`.

Es wird derzeit ein möglichst breites Spektrum an Daten in hoher zeitlicher Auflösung erfasst, um in einem Prozess der Datenexploration auf Zusammenhänge schließen zu können, die zunächst nicht offensichtlich sind bzw. um Vermutungen zu validieren. Derzeit werden über 300 Kennzahlen alle 10 s auf 14 Servern aus 9 Kundenprojekten abgetastet. Diese Daten werden gespeichert und außerdem unmittelbar weiterverarbeitet. So findet etwa ein Downsampling für alle genannten 300 Kennzahlen statt. Dabei werden die zeitliche Auflösung unter Verwendung verschiedener Aggregatfunktionen auf Zeitfenster unterschiedlicher Größe reduziert und die Ergebnisse gespeichert. Andere Analysefunktionen quantisieren die Werte hinsichtlich ihrer Zugehörigkeit zu Statusklassen (etwa optimal, normal, kritisch) und speichern die Ergebnisse ebenfalls. So entstehen sehr schnell große Datenmengen. Derzeit enthält der Datenspeicher etwa 40 GB Daten und wir beobachten bei der aktuellen Anzahl beobachteter Werte einen Zuwachs von etwa 1 GB Daten pro Woche. Auf Basis der erhobenen Daten müssen zeitkritische Analysen wie etwa eine Ausreißererkennung oder die Erkennung kritischer Muster nahezu in Echtzeit erfolgen, um Kunden ein rechtzeitiges Eingreifen zu ermöglichen. Weiterhin soll eine Vorhersage zukünftiger Werte frühzeitig kritische Entwicklungen aufzeigen. Die Herausforderung im Projekt ist die Bewältigung des großen Datenvolumens unter Gewährleistung einer echtzeitnahen Bearbeitung durch Analysefunktionen.

3. ZEITREIHENANALYSE UND DATENBANKEN

Im Rahmen der Evaluierung von für das Projekt geeigneter Software haben wir verschiedene Ansätze zur Datenstromverarbeitung und der Analyse und Verwaltung von Zeitreihen untersucht. Ziel war die Verwendung frei verfügbarer Software die zudem auf im Unternehmen vorhandener technischer Expertise basiert.

3.1 Data Stream Management Systems

Die Verarbeitung kontinuierlicher Datenströme stellt einen Aspekt unseres Projektes dar. Datenstromverarbeitende Systeme bieten hierzu die Möglichkeit kontinuierliche Anfragen auf in temporäre Relationen umgewandelte Datenströme zu formulieren. Dies kann etwa mit Operatoren der im Projekt Stream[1] entwickelten an SQL angelehnten Continuous Query Language[2] erfolgen. Sollen nun komplexere Muster in Datenströmen erkannt werden, spricht man auch von der Verarbeitung komplexer Ereignisse. Im Kontext unseres Projektes entspricht so ein Muster etwa dem Anstieg der Aufrufe einer Seite aufgrund einer Marketingaktion, welcher eine höhere Systemauslastung zur Folge hat (cpu-usage), was sich wiederum in steigenden time-to-first-byte-Werten niederschlägt und in einem kritischen Bereich zur Benachrichtigung oder gar zur automatischen Aufstockung der verfügbaren Ressourcen führen soll. Complex Event Processing Systems wie Esper[5] bieten die Möglichkeit Anfragen nach solchen Mustern auf Datenströme zu formulieren und entsprechende Reaktionen zu implementieren. Da etwa Esper als eines der wenigen frei verfügbaren und für den produktiven Einsatz geeigneten Systeme, in Java und .net implementiert ist, entsprechende Entwicklungskapazitäten jedoch nicht im Unternehmen zur Verfügung stehen, wird im Projekt keines der erwähnten DSMS oder CEPS zum Ein-

satz kommen. Deren Architektur diente jedoch zur Orientierung bei der Entwicklung eines eigenen mit im Unternehmen eingesetzten Techniken (etwa node.js[6], RabbitMQ[7], MongoDB[8], u.a.) Systems für PageBeat.

3.2 Werkzeuge zur Datenanalyse

Zur statistischen Auswertung der Daten im Projekt werden Werkzeuge benötigt, die es ohne großen Implementierungsaufwand ermöglichen verschiedene Verfahren auf die erhobenen Daten anzuwenden und auf ihre Eignung hin zu untersuchen. Hierfür stehen verschiedene mathematische Werkzeuge zur Verfügung. Kommerzielle Produkte sind etwa die bereits erwähnten Matlab oder SPSS. Im Bereich frei verfügbarer Software kann man auf WEKA und vor allem R zurückgreifen. Besonders R ist sehr weit verbreitet und wird von einer großen Entwicklergemeinde getragen. Dadurch sind für R bereits eine Vielzahl von Verfahren zur Datenaufbereitung und deren statistischer Analyse bis hin zur entsprechenden Visualisierung implementiert. Gerade in Bezug auf die Analyse von Zeitreihen ist R aufgrund vielfältiger verfügbarer Pakete zur Zeitreihenanalyse gegenüber WEKA die geeignetere Wahl. Mit RStudio[9] steht außerdem eine komfortable Entwicklungsumgebung zur Verfügung. Weiterhin können mit dem Web Framework Shiny[10] schnell R Anwendungen im Web bereit gestellt werden und unterstützt somit eine zügige Anwendungsentwicklung. Somit stellt R mit den zugehörigen Erweiterungen die für das Projekt geeignete Umgebung zur Evaluierung von Datenanalyseverfahren und zur Datenexploration dar. Im weiteren Verlauf des Projektes und in der Überführung in ein produktives System wird die Datenanalyse, etwa die Berechnung von Vorhersagen, innerhalb von node.js reimplementiert.

3.3 Datenbankunterstützung

Klassische objektrelationale DBMS wie Oracle[11], IBM Informix[12] oder PostgreSQL[13] unterstützen in unterschiedlichem Umfang die Speicherung, Anfrage und Auswertung von Zeitreihen. PostgreSQL ermöglicht bswp. die Verwendung von Fensterfunktionen etwa zur Berechnung von Aggregatwerten für entsprechende Zeitabschnitte. Die IBM Informix TimeSeries Solution[3] stellt Container zur Speicherung von Zeitreihendaten zur Verfügung, wodurch der Speicherplatzbedarf optimiert, die Anfragegeschwindigkeit erhöht sowie die Komplexität der Anfragen reduziert werden sollen. Oracle unterstützt nicht nur die Speicherung und Anfrage von Zeitreihen, sondern integriert darüber hinaus umfassende statistische Analysefunktionalität mittels Oracle R Technologies[4]. Dabei hat der R-Anwendungsentwickler die Möglichkeit Oracle Data Frames zu verwenden, um Datenlokalität zu erreichen. Dabei wird der Code in der Oracle-Umgebung ausgeführt, dort wo die Daten liegen und nicht umgekehrt. Außerdem erfolgt so ein transparenter Zugriff auf die Daten und Aspekte der Skalierung werden durch das DBMS abgewickelt.

Neben den klassischen ORDBMS existieren eine Vielzahl von auf Zeitserien spezialisierte Datenbanken wie OpenTSDB[14], KairosDB[15], RRDB[16]. Dabei handelt es sich jeweils um einen auf Schreibzugriffe optimierten Datenspeicher in Form einer schemalosen Datenbank und darauf zugreifende Anfrage-, Analyse- und Visualisierungsfunktionalität. Man sollte sie deshalb vielmehr als Ereignis-Verarbeitungs- oder Monitoring-Systeme bezeichnen. Neben den bisher genannten Zeitseriendatenbanken ist uns bei der Recherche von für das Projekt geeigneter Software InfluxDB[17] aufgefallen. InfluxDB verwendet Googles auf Log-structured merge-trees basierenden key-value Store LevelDB[18] und setzt somit auf eine hohen Durchsatz bzgl. Schreiboperationen. Einen Nachteil hingegen stellen langwierige Löschoperationen ganzer nicht mehr benötigter Zeitbereiche dar. Die einzelnen Zeitreihen werden bei der Speicherung sequenziell in sogenannte Shards unterteilt, wobei jeder Shard in einer einzelnen Datenbank gespeichert wird. Eine vorausschauenden Einrichtung verschiedener Shard-Spaces (4 Stunden, 1 Tag, 1 Woche etc.) ermöglicht es, das langsame Löschen von Zeitbereichen durch das einfache Löschen ganzer Shards also ganzer Datenbanken (drop database) zu kompensieren. Eine verteilte Speicherung der Shards auf verschiedenen Rechnerknoten die wiederum in verschiedenen Clustern organisiert sein können, ermöglicht eine Verteilung der Daten, die falls gewünscht auch redundant mittels Replikation auf verschiedene Knoten erfolgen kann. Die Verteilung der Daten auf verschiedene Rechnerknoten ermöglicht es auch die Berechnung von Aggregaten über Zeitfenster die unterhalb der Shardgröße liegen, zu verteilen und somit Lokalität der Daten und einen Performance-Vorteil zu erreichen. Auch hier ist es sinnvoll Shardgrößen vorausschauend zu planen. Die Anfragen an InfluxDB können mittels einer SQL-ähnlichen Anfragesprache über eine http-Schnittstelle formuliert werden. Es werden verschiedene Aggregatfunktionen bereitgestellt, die eine Ausgabe bspw. gruppiert nach Zeitintervallen für einen gesamten Zeitbereich erzeugen, wobei die Verwendung Regulärer Ausdrücke unterstützt wird:

```
select median(used) from /cpu\.*/
where time > now() - 4h group by time(5m)
```

Hier wird der Median des „used“-Wertes für alle 5-Minuten-Fenster der letzten 4 Stunden für alle CPUs berechnet und ausgegeben. Neben normalen Anfragen können auch sogenannte Continuous Queries eingerichtet werden, die etwa das einfache Downsampling von Messdaten ermöglichen:

[6] node.js - a cross-platform runtime environment for server-side and networking applications. `http://nodejs.org/`.
[7] RabbitMQ - Messaging that just works. `http://www.rabbitmq.com`.
[8] MongoDB - An open-source document database. `http://www.mongodb.org/`.
[9] RStudio - open source and enterprise-ready professional software for the R statistical computing environment. `http://www.rstudio.com`.
[10] Shiny - A web application framework for R. `http://shiny.rstudio.com`.
[11] Oracle. `http://www.oracle.com`.
[12] IBM Informix. `http://www-01.ibm.com/software/data/informix/`.
[13] PostgreSQL. `http://www.postgresql.org/`.
[14] OpenTSDB - Scalable Time Series Database. `http://opentsdb.net/`.
[15] KairosDB - Fast Scalable Time Series Database. `https://code.google.com/p/kairosdb/`.
[16] RRDB - Round Robin Database. `http://oss.oetiker.ch/rrdtool/`.
[17] InfluxDB - An open-source distributed time series database with no external dependencies. `http://influxdb.com/`.
[18] LevelDB - A fast and lightweight key-value database library by Google. `http://code.google.com/p/leveldb/`.

```
select count(name) from clicks
group by time(1h) into clicks.count.1h
```

InfluxDB befindet sich noch in einem frühen Stadium der Entwicklung und wird ständig weiterentwickelt. So ist etwa angekündigt, dass zukünftig bspw. das Speichern von Metadaten zu Zeitreihen (Einheiten, Abtastrate, etc.) oder auch die Implementierung nutzerdefinierter Aggregatfunktionen ermöglicht wird. InfluxDB ist ein für unsere Anwendung vielversprechendes Werkzeug, wobei jedoch abzuwarten bleibt, inwiefern es sich für den produktiven Einsatz eignet. Aus diesem Grund wird derzeit zusätzlich zu InfluxDB, MongoDB parallel als im Unternehmen bewährter Datenspeicher verwendet.

4. LÖSUNG IN PAGEBEAT

Im Projekt Pagebeat wurden verschiedene Lösungsansätze getestet, wobei die Praktikabilität beim Einsatz im Unternehmen, die schnelle Umsetzbarkeit sowie die freie Verfügbarkeit der eingesetzten Werkzeuge die entscheidende Rolle spielten.

4.1 Datenfluss

Der Datenfluss innerhalb der Gesamtarchitektur ist in Abbildung 1 dargestellt. Die Messdaten werden von einer Drohne[19] sowie Clientsimulatoren und Lasttestservern in äquidistanten Zeitabschnitten (meist 10 s) ermittelt. Die erhobenen Daten werden einem Loggingdienst per REST-Schnittstelle zur Verfügung gestellt und reihen sich in die Warteschlange eines Nachrichtenservers ein. Von dort aus werden sie ihrer Signatur entsprechend durch registrierte Analyse- bzw. Interpretationsprozesse verarbeitet, wobei die Validierung der eintreffenden Daten sowie die Zuordnung zu registrierten Analysefunktionen mittels einer Wissensbasis erfolgt. Ergebnisse werden wiederum als Nachricht zur Verfügung gestellt und falls vorgesehen persistent gespeichert. So in die Nachrichtenschlange gekommene Ergebnisse können nun weitere Analysen bzw. Interpretationen oder die Auslösung einer Benachrichtigung zur Folge haben. Der Daten Explorer ermöglicht eine Sichtung von Rohdaten und bereits in PageBeat integrierten Analyseergebnissen sowie Tests für zukünftige Analysefunktionen.

4.2 Wissensbasis

Die Wissenbasis bildet die Grundlage für die modular aufgebauten Analyse- und Interpretationsprozesse. Die Abbildung 2 dargestellten „ParameterValues“ repräsentieren die Messdaten und deren Eigenschaften wie Name, Beschreibung oder Einheit. ParameterValues können zu logischen Gruppen (Parameters) zusammengefasst werden (wie z.B. die ParameterValues: „system“, „load“, „iowait“ und „max“ zum Parameter „cpu“). Parameter sind mit Visualisierungskomponenten und Kundendaten sowie mit Analysen und Interpretationen verknüpft. Analysen und Interpretationen sind modular aufgebaut und bestehen jeweils aus Eingangs- und Ausgangsdaten (ParameterValues) sowie aus Verweisen auf den Programmcode. Weiterhin sind ihnen spezielle Methodenparameter zugeordnet. Hierbei handelt es sich etwa um Start und Ende eines Zeitfensters, Schwellenwerte oder andere Modellparameter. Die Wissensbasis ist mittels eines relationalem Schemas in MySQL abgebildet.

[19] Auf dem zu beobachtenden System installierter Agent zur Datenerhebung.

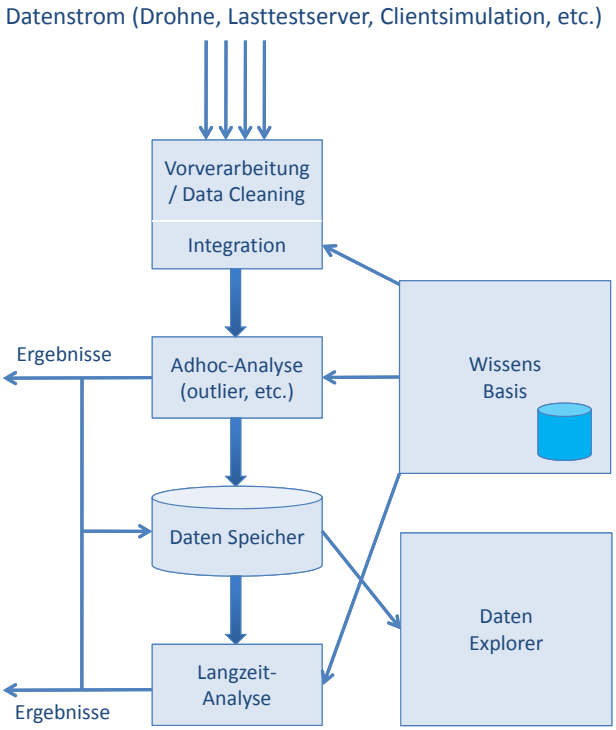

Abbildung 1: Datenfluss

4.3 Speicherung der Zeitreihen

Die Speicherung der Messdaten sowie Analyse- und Interpretationsergebnisse erfolgt zum Einen in der im Unternehmen bewährten, auf hochfrequente Schreibvorgänge optimierten schemafreien Datenbank MongoDB. Zum Anderen setzen wir mittlerweile parallel zu MongoDB auf InfluxDB. So kann z.B. über die in InluxDB zur Verfügung stehenden Continious Queries ein automatisches Downsampling und somit eine Datenreduktion der im 10 Sekunden Takt erhobenen Daten erfolgen. Das Downsampling erfolgt derzeit durch die Berechnung der Mittelwerte von Zeitfenstern einer Länge von 1 Minute bis hin zu einem Tag und generiert somit automatisch unterschiedliche zeitliche Auflösungen für alle Messwerte. Außerdem stellt die SQL ähnliche Anfragesprache von InfluxDB eine Vielzahl von für die statistische Auswertung hilfreichen Aggregatfunktionen (min, max, mean, median, stddev, percentile, histogramm, etc.) zur Verfügung. Weiterhin soll es zukünftig möglich sein benutzerdefinierte Funktionen mit eigener Analysefunktionalität (etwa Autokorrelation, Kreuzkorrelation, Vorhersage, etc.) auf Datenbankebene umzusetzen oder auch das automatische Zusammenführen verschiedener Zeitserien anhand eines Timestamp-Attributs durchzuführen. Dies würde schon auf Datenbankebene eine zeitreihenübergreifende Analyse (bspw. Korrelation) unterstützen und senkt den Reimplentierungsaufwand von R Funktionalität aus der Datenexplorationsphase. Da herkömmliche Datenbanken nicht die hohe Performance bzgl. Schreibzugriffen erreichen und kaum auf Zeitreihen spezialisierte Anfragen unterstützen, scheint InfluxDB ein geeigneter Kandidat für den Einsatz innerhalb PageBeats zu sein.

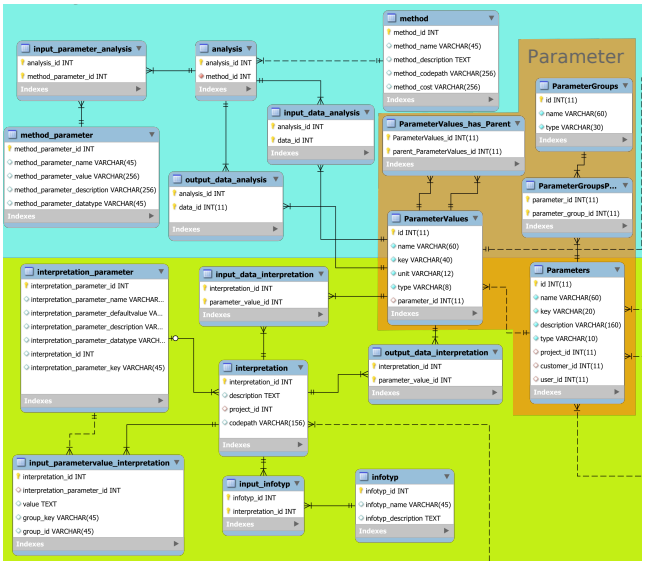

Abbildung 2: Ausschnitt Schema Wissensbasis

4.4 Datenexploration

Die Datenexploration soll dazu dienen, Administratoren und auch Endnutzern die Möglichkeit zu geben, die für sie relevanten Daten mit den richtigen Werkzeugen zu analysieren. Während der Entwicklung nutzen wir die Datenexploration als Werkzeug zur Ermittlung relevanter Analysemethoden und zur Evaluierung sowie Visualisierung der Datenströme. Abbildung 3 zeigt eine einfache Nutzerschnittstelle umgesetzt mit Shiny zur Datenauswertung mittels R mit Zugriff auf unterschiedliche Datenbanken, InfluxDB und MongoDB. Verschiedene Parameter zur Auswahl des Zeitraumes, der Analysefunktion und deren Parameter sowie Visualisierungsparameter.
Hier sind durchschnittliche CPU-Nutzung und durchschnittliche Plattenzugriffszeiten aus einer Auswahl aus 10 Zeitserien dargestellt. Mittels unterem Interaktionselement lassen sich Intervalle selektieren und die Granularität anpassen. Mit ähnlichen Visualisierungsmethoden lassen sich auch Autokorrelationsanalysen visualisieren, siehe Abbildung 4.

4.5 Analyse und Interpretation

Analysen sind Basisoperationen wie die Berechnung von Mittelwert, Median, Standardabweichung, Autokorrelation u.a. deren Ergebnisse falls nötig persistent gespeichert werden oder direkt anderen Verarbeitungsschritten als Eingabe übergeben werden können. Die Spezifizierung der Analysefunktionen erfolgt in der Wissensbasis, die eigentliche Implementierung ist möglichst nahe an den zu analysierenden Daten, wenn möglich unter Verwendung von Aggregat- oder benutzerdefinierten Funktionen des Datenbanksystems, umzusetzen. Wissensbasis und Analyse sind hierzu mittels eines „method_codepath“ verknüpft.
Interpretationen funktionieren analog zur Analyse bilden jedoch Berechnungsvorschriften etwa für den Gesamtindex (Pagebeat-Faktor) des Systems bzw. einzelner Teilsysteme ab, in dem sie z.B. Analyseergebnisse einzelner Zeitreihen gewichtet zusammenführen. Weiterhin besitzen Interpretationen einen Infotyp, welcher der nutzerspezifischen Aufbereitung von Ergebnissen dient. Abbildung 5 zeigt etwa die Darstellung aggregierter Parameter in Ampelform (rot = kritisch, gelb = Warnung, grün = normal, blau = optimal) was schnell einen Eindruck über den Zustand verschiedener Systemparameter ermöglicht.

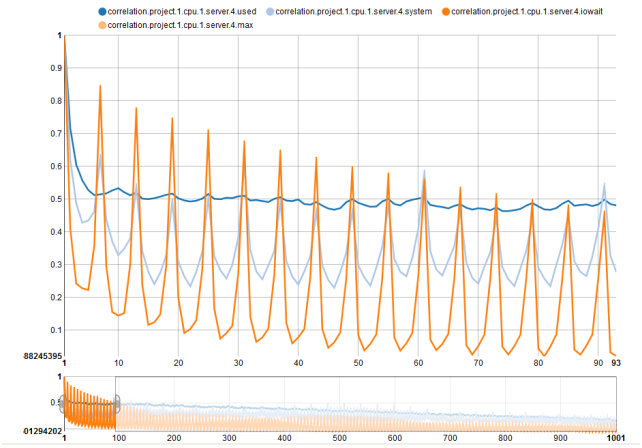

Abbildung 4: Autokorrelation

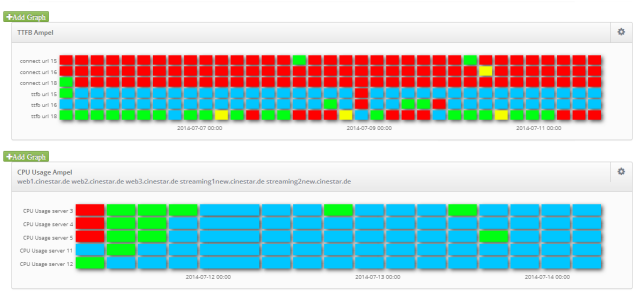

Abbildung 5: Ampel

Analysefunktionalität die über Aggregationen auf Datenbankebene hinausgehen wird von uns in einer Experimentalumgebung umgesetzt und evaluiert. Diese basiert auf R. So stehen eine Vielzahl statistischer Analysemethoden und Methoden zur Aufbereitung komplexer Datenstrukturen in Form von R Paketen zur Verfügung. Darüber hinaus ermöglicht das R-Paket „Shiny Server“ die komfortable Bereitstellung von R Funktionalität für das Web. Ein wesentlicher Teil unser Experimentalumgebung ist der Pagebeat Data Explorer (siehe Abbildung 3). Dieser basiert auf den genannten Techniken und ermöglicht die Sichtung der erfassten Rohdaten oder das „Spielen“ mit Analysemethoden und Vorhersagemodellen.

5. ZUSAMMENFASSUNG UND AUSBLICK

Pagebeat ist ein Projekt, bei dem es insbesondere auf eine performante Speicherung und schnelle Adhoc-Auswertung der Daten ankommt. Dazu wurden verschiedene Lösungsansätze betrachtet und die favorisierte Lösung auf Basis von InfluxDB und R beschrieben.
Die konzeptionelle Phase ist abgeschlossen, die Projektinfrastruktur umgesetzt und erste Analysemethoden wie Aus-

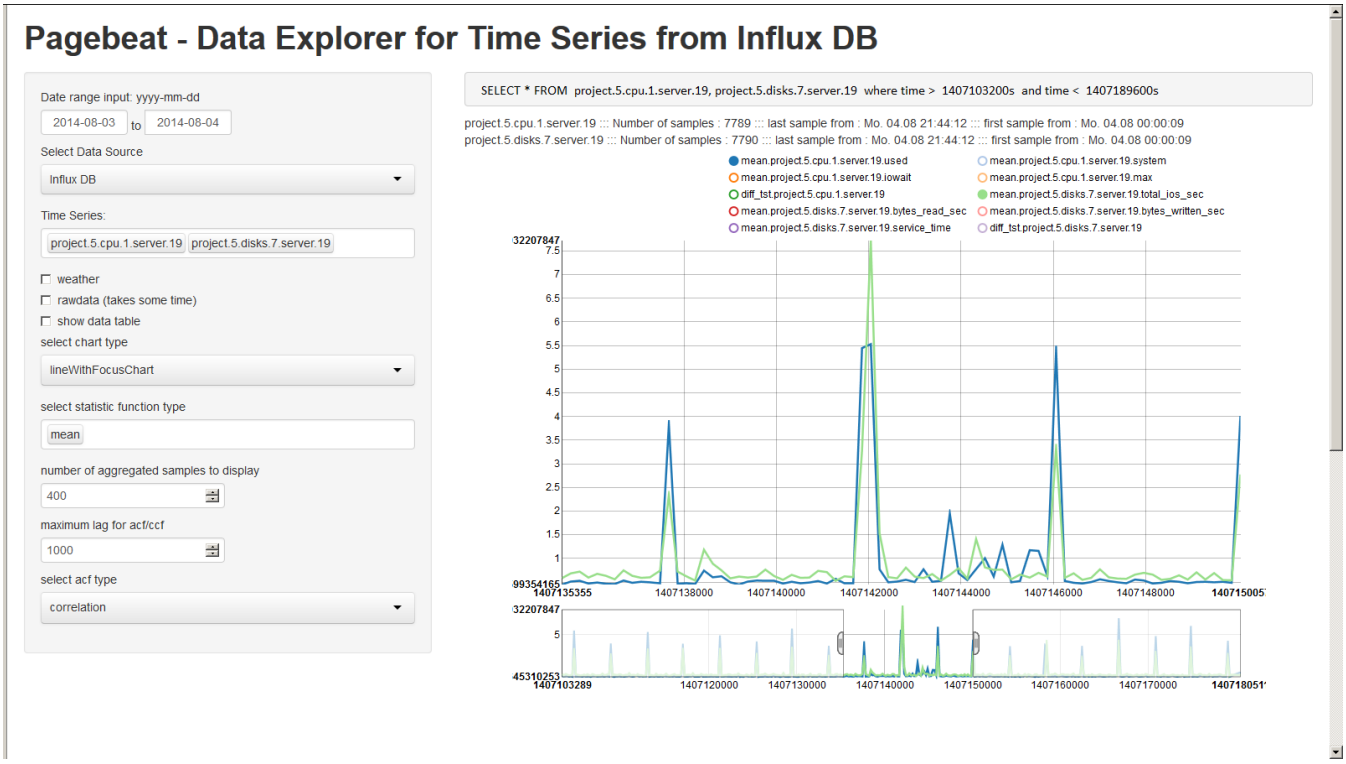

Abbildung 3: Daten

reißererkennung oder Autokorrelation wurden ausprobiert. Derzeit beschäftigen wir uns mit den Möglichkeiten einer Vorhersage von Zeitreihenwerten. Dazu werden Ergebnisse der Autokorrelationsanalyse zur Identifikation von Abhängigkeiten innerhalb von Zeitreihen verwendet um die Qualität von Vorhersagen abschätzen zu können. Weiterhin ist geplant Analysen näher an der Datenbank auszuführen um Datenlokalität zu unterstützen.

6. REFERENCES

[1] A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava, and J. Widom. Stream: The stanford data stream management system. Technical Report 2004-20, Stanford InfoLab, 2004.

[2] A. Arasu, S. Babu, and J. Widom. The cql continuous query language: Semantic foundations and query execution. Technical Report 2003-67, Stanford InfoLab, 2003.

[3] K. Chinda and R. Vijay. Informix timeseries solution. `http://www.ibm.com/developerworks/data/library/techarticle/dm-1203timeseries`, 2012.

[4] O. Corporation. R technologies from oracle. `http://www.oracle.com/technetwork/topics/bigdata/r-offerings-1566363.html`, 2014.

[5] EsperTech. Esper. `http://esper.codehaus.org`, 2014.

[6] U. Fischer, L. Dannecker, L. Siksnys, F. Rosenthal, M. Boehm, and W. Lehner. Towards integrated data analytics: Time series forecasting in dbms. *Datenbank-Spektrum*, 13(1):45–53, 2013.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.