GeoVisual Analytics Platform for large-scale Multi-Agent Simulations

Janus Dybulla
janus-dybulla@haw-hamburg.de
Hamburg University of Applied Sciences,
Dept. Computer Science,
Hauptseminar WS 16/17

Abstract

Multi-Agent Systems produce massive amounts of spatio-temporal data that is created by complex models and large-scale simulations. These enormous datasets require special treatment that cannot be satisfied with traditional database systems that are used to store, manage, and access data. Moreover, the tremendous amount of data challenges simulation modelers to interpret and analyze this results by interactive exploration of a simulation. Generic and near real-time performant visual-interactive interfaces based on multi-agent simulations are very rare. Most analytical systems are specifically designed for a particular domain.

In the on-going study projects I will accomplish a distributed computing cloud platform based on the hadoop ecosystem as a viable solution for large-scale analytics for the MSaaS system MARS 2.0. Performance characteristics of distributed data streaming platforms and geo-spatial database systems are considered. Motivated by the idea of Visual Analytics, a web-based extendable visualization dashboard is build up on the AaaS cloud infrastructure enabling decision makers to evaluate different simulation scenarios. It provides a straightforward overview of the simulation output and allows the modeler to build sql queries and explore the dataset through interactively linked views. During simulation runs, and additionally on demand, the modeler can discover potentially interesting patterns in the datasets by selecting and defining different graph and map types, like scalable heat maps. The platform should offer the modeler an environment to process their own analytical tasks written in their favourite programming language and compute it on distributed nodes. Projects like Zeppelin and Jupyter still lack of real time data analysis with dashboard capabilities. Especially in the context of large-scale agend-based simulations including geo-spatial multivariate data. The proposed platform will operate as a demonstrator and as the founding stone for further investigations regarding the master thesis outline.

Schlüsselwörter: visual analytics, model-agent based simulation, big data, data analysis, information visualisation, geospatial, temporal, geovisual analytics

1 Einleitung

Die Entwicklung von Systemen für computergestützte Simulationen begann mit reinen, für sich stehenden Simulationsengines. In der Forschung steht mittlerweile die vollständige Unterstützung kompletter Prozesse mit einer Simulationsplattform im Mittelpunkt [RHWU14]. Die jeweiligen Simulations-Prozesse können dabei je nach Zielsetzung verschiedene Aufgaben erfüllen, wie etwa die geführte Erstellung von Simulationsmodellen, Kriterien zur Steuerung von Experimenten sowie statistische Auswertungen oder die interaktive Exploration der geographischen Verteilung von Schlüsselakteuren. All diese Aufgaben erfordern eine Datensammlung, Datenanalyse sowie überwiegend auch eine Datenspeicherung [SMU14]. Zunehmende Komplexität der Simulationsmodelle von Multi-Agenten-Simulationen (MAS) mit mehreren Millionen Agenten haben auf verteilten, skalierbaren Simulationsumgebungen zu höherem Datendurchsatz, größerem Datenvolumen und vielfältigen Datentypen geführt [ZVCK16]. Bei besonders rechenintensiven Simulationen mit langen Laufzeiten wird die Ausführung vieler Simulationen vermindert, indem möglichst viele Daten mit so wenigen Durchläufen wie möglich aufgezeichnet werden. Der jeweils zur Datenanalyse und -speicherung gewählte Ansatz wirkt sich daher insgesamt auf die Leistung einer Simulationsstudie aus. Mit traditionellen Datenbanksystemen muss erst eine Speicherung der Daten erfolgen, bevor eine darauf bezogene Abfrage möglich ist, wodurch diese für Echtzeit Datenanalysen während der Datenentstehung ungeeignet sind [SMU14].

Für die Datenverarbeitung in Echtzeit wurden daher spezielle sogenannte Data Stream Management Systeme entwickelt, welche mit kontinuierlich und potentiell unendlich ablaufenden Datenströmen von oft strukturierten Daten arbeiten, von denen nur ein kleiner Teil gespeichert wird. Mehrere Datenströme können dabei gefiltert, aggregiert, kombiniert, miteinander verrechnet und überwacht werden [HZEF16].

Bei anderen Ansätzen, die mit komplexen, analytischen Anfragen einhergehen, können bestimmte Analysemethoden „in-situ“ in der Simulationsumgebung bei der Datenerstehung ansetzen. Die direkte Kopplung kann hierbei simultan oder blockierend zur Simulationsausführung geschehen, um beispielsweise Stopbedingungen oder Parameterräume der Simulation zu überprüfen [BAA+16]. Die stärkere Verflechtung von Daten und Analysemethoden wird besonders zur Unterstützung von Entscheidungsprozessen als wichtig erachtet [SMU14]. Aus diesem Grund gibt es jüngst im Bereich der Datenbankentwicklung Bestrebungen, Simulationstechniken direkt in die Datenbanken zu integrieren. Auch wenn es sehr erstrebenswert ist, Datenmanagement und Simulationstechniken näher aneinander zu rücken, so birgt diese Architekturentscheidung besonders in verteilten und komponentengestützten Simulationsumgebungen Probleme bei der Skalierung. Die Herangehensweise für die an der HAW entwickelte cloud-basierte Multi-Agenten Simulationsplattform MARS [HATCea16] wird in Kap. 4 beschrieben. Visualisierungstools haben das Potential für ein besseres Verständnis von Daten

zu sorgen, in dem sie ihren Anwendern mehr Verarbeitungsressourcen zur Verfügung stellen, die Suche nach Informationen erleichtern, Muster automatisch erkennen und Mechanismen für Schlussfolgerungen anbieten [CAD+14]. Visualisierungen befähigen Menschen zu einem produktiven Umgang mit Big Data, der noch über das Interpretationsvermögen von Maschinen hinausgeht. Sie bringen aber auch ein gewisses Risiko der Irreführung von Anwendern durch Fehlinterpretationen oder kognitiver Überlastung mit [Dyb16]. Im Vergleich dazu erkennen Verfahren und Prozesse aus der Visual Analytics ebenfalls die menschlichen Einschränkungen an und kombinieren die visuelle Intelligenz von Menschen mit der künstlichen Intelligenz computergestützter Werkzeuge [AA13]. Sie fördern damit die Kommunikation von Hypothesen, Ergebnissen und Ideen. Entwickler von Visualisierungstools stoßen bei der Gestaltung neuer Werkzeuge für Simulationsdaten daher auf einige Hürden und das Ausprobieren neuer Ansätze und Ideen führt zu einer wachsenden Zahl individueller Tools. Dass viele dieser Tools für den Eigenbedarf entwickelt werden, schränkt ihren praktischen Nutzen ein. Und trotz vieler Fortschritte in der Informatik greifen viele Anwender im professionellen Umfeld auf veraltete Visualisierungstools und Datenmanagementsysteme zurück, die dem aktuellen Bedarf nicht mehr gerecht werden [CAD+14].

Der hier vorgestellte Ansatz hebt die Datennutzung in cloudbasierten Simulationen hervor. Es werden zu erst Informationsbedürfnisse und Nutzeranforderungen ermittelt, Funktionen und Systemarchitekturen bestehender Tools beschrieben und Erwägungen für Usability und komponenten-basierte Erweiterungen mit einbezogen. Darauf aufbauend wird das konkret geplante Vorgehen und weitere Schritte umrissen sowie Evaluierungskriterien beschrieben. Abschließend werden einige der zu erwartenden Herausforderungen und Chancen für die Entwicklung neuartiger Visual Analytics-Tools dargelegt.

1.1 Hintergründe und Anforderungen

Multi-Agenten Simulationen (MAS) bestehen üblicherweise aus vielen autonom interagierenden Agenten und können formal durch zustandslose und zustandsverändernde Funktionen für diese Agenten und eine Interaktions-Topologie für das gesamte System beschrieben werden [GSL+11]. Die Agenten bewegen sich hierbei stets in einer virtuellen Umwelt bzw. einem geographischen Raum. Für das hier vorgestellte integrierte Datenmanagement- und Visualisierungs-System wird von einem geospatialen Environment mit GPS-Koordinaten ausgegangen. Viele Abfragen erfordern daher geometrische Berechnungen, wie etwa aufwändige Berechnungen für sich ständig verändernde, räumliche Beziehungen. Da viele Agenten kontinuierlich ihren aktualisierten Standort registrieren, benötigt die Datenspeicherlösung eine hohe Schreibgeschwindigkeit um mit dem Datenvolumen zurechtzukommen. Die Datenbank sollte gut skalierbar und fehlertolerant sein sowie hohe Verfügbarkeit und eine geringe Latenz beim Verarbeiten großer Datenmengen aufweisen. Dem drängenden Bedarf große, geospatiale Daten zu verwalten und zu analysieren, steht leider ein Mangel spezialisierter Sys-

teme, Techniken und Algorithmen für diese Datenarbeit gegenüber. Big Data wird beispielsweise durch eine Vielzahl an Map-Reduce-ähnlichen Systemen und Cloudinfrastrukturen unterstützt (z.B. Haddop, Hive, HBase, Impala, Drill und Storm) [EM15]. Keines dieser Systeme bzw. keine der genannten Infrastrukturen unterstützt explizit räumliche Daten oder Raum-Zeit Daten.

Was neue Designs und Visualisierungsparadigmen betrifft, gibt es bereits zahlreiche alternative Ansätze, von denen viele aber noch nicht empirisch validiert werden konnten. Am häufigsten findet man darunter untereinander mehrfach verlinkte Ansichten (multiple linked views), bei denen Methoden wie Brushing oder Verlinkung es dem Anwender ermöglichen, mit mehreren Visualisierungen gleichzeitig zu arbeiten. Als Ergänzungen wurden auch Ansätze zur Zusammenfassung, Clusterbildung und Hervorhebung vorgeschlagen [LDC+16]. Mit dem Vorstellungsvermögen arbeitende Ansätze, wie Fokus- und Kontext-Visualisierungen, könnten interessante neue Möglichkeiten bieten, da sie eine Reduktion der Informationsdarstellung anstreben [LDC+16]. Visual Analytics-Tools sollten leicht verfügbar und bedienbar sein, um Wissenschaftlern aller Fachbereiche, Spezialisten und auch der allgemeinen Öffentlichkeit die Erkundung, Analyse und Übermittlung von Daten in einer kollaborativen Umgebung zu ermöglichen.

Auf eine explizite Bewertung und Kalibrierung eines Simulationsmodells durch „Fitness“-Funktionen oder Output-Indikatoren, die im Grundseminar vorgestellt wurden, wird hier zugunsten einer allgemeinen GeoVisual Analytics-Plattform verzichtet. Empirische Daten und Experten-Knowhow sollten dennoch zur Validierung einbezogen werden können [Dyb16].

2 Rückblick: Prototyp

Lange Zeit gab es keine verlässliche Möglichkeit Simulationsergebnisse der MARS-Plattform auszuwerten. Um dennoch Modellexperimente durchführen zu können, wurde zu Beginn eine leichtgewichtige Architektur mit wenigen Komponenten gewählt. Dies erleichterte zwar die schnelle Inbetriebnahme und Wartung, skalierte aber sichtlich schlecht. Zum Beispiel wurde anfangs eine MongoDB als Datenbank zum Speichern aller Eingaben und Ergebnisse der Simulationen genutzt, da diese auch über geospatiale Abfragen verfügt. Dies hatte den Vorteil, dass alle Ursprungsdaten in der gleichen Dokumentendatenbank gespeichert werden konnten. Mit einer zunehmenden Anzahl an Simulationen erwies sich die MongoDB aber aufgrund beispielsweise nicht linearer Durchsatzgeschwindigkeiten für das Abspeichern und Lesen großer Dateien als ungeeignet. Eine eigens geschaffene Weboberfläche bietet direkten Zugang zu den Daten in Form von einigen Visualisierungen, die Ansätze der Visual Analytics implementieren und in Abb. 1 zu sehen sind. Die Echtzeit-Analyse bzw. Sub-Second-Anfragen sind hierbei nur bedingt durchführbar, da die Daten mit einer Stapelverarbeitung zu festen Intervallen aus der Datenbank extrahiert werden. Eine Zeitleiste mit den jeweiligen Agentenpopulationen kann zur temporalen Filterung genutzt werden. Um ein

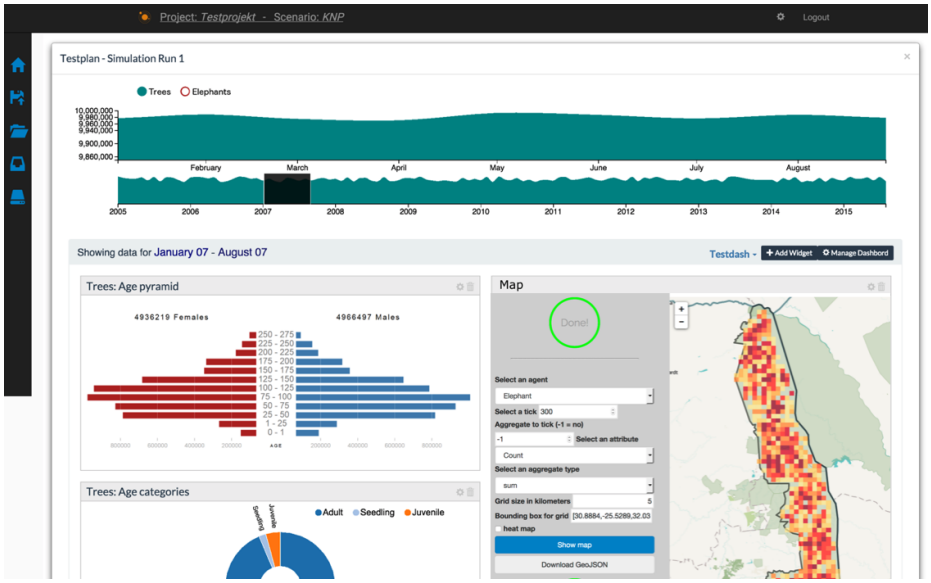


Abb. 1. Screenshot von der Weboberfläche des integrierten MARS Prototypen.

Überplotten zu vermeiden, enthält diese gesampelte und aggregierte Daten. Die einzelnen Ansichten in dem Dashboard sind mit dem ausgewählten Zeitraum verlinkt. Eine Kartenansicht gibt Aufschluss über die Verbreitung, Verteilung und Bewegung einzelner Agenten. Spatiale Filterung sowie Clustering in Form einer Heatmap nach [LJH13] wird unterstützt. Eine schnell zu berechnenden Repräsentation des geometrischen Clusterings, die sogenannte „binned aggregation“, gruppiert Agenten in wählbare, vordefinierte Kästen [PBH+15]. Weiterhin ist die Veranschaulichung einzelner Agentenbewegungen durch Trajektoriensegmente gegeben, vgl. dazu [YG16]. Jedes Segment stellt eine definierte Zeitspanne dar und zeigt einzelne Attributwerte des Agenten durch verschieden kolorierte Segmente auf. Für weitere Analysezwecke kann der Benutzer Teilergebnisse herunterladen oder muss direkt mit der Datenbank kommunizieren. Folglich sind einige der in Kap. 1.1 genannten Anforderungen nicht bzw. nur sehr schwer mit dem vorhandenen Prototypen umsetzbar. In der Literatur wurden weitere Ansätze vorgeschlagen, um Vorteile technologischer Neuerungen wie Cloud-Computing, Datenbankindexing und -abfragen zu nutzen.

3 Cloud-basierte Plattformen für Multi-Agenten-Simulationen

Nach wie vor ist die Datenhaltung der Simulationsergebnisse mittels Kommaseparierten-, XML- oder applikationsspezifischen Binärdateiformaten weit ver-

breitet [ZVCK16]. Für Simulationen mit nur ein paar Tausend Agenten oder wenigen Schreibvorgängen mag dies noch akzeptabel sein. Oft werden bei größeren Simulationen daher nur aggregierte und für eine bestimmte Visualisierung wichtige Daten gespeichert [YYH+17]. Ein vollumfänglicher und universeller Zugriff auf die Rohdaten ist dadurch aber ausgeschlossen. Die beschränkte Datentransferrate zu einer meist zentralisierten Recheneinheit und der für das Veröffentlichen aller Informationen erforderliche Overhead wirkt sich entweder negativ auf die Simulations-Performance aus oder macht längere Cool-Down-Phasen während Simulationen erforderlich, in denen die Daten ausgewertet und an einzelne Analyseeinheiten übertragen werden [ZVCK16].

Li et al. nutzen mit ihrem System 4D-SAS die Möglichkeiten des Cloud-Computings durch verteilte Datenverarbeitung in leichtgewichtigen Linuxcontainern (hier: Docker) aus. Als Simulationsengine wurde die für Multi-Agenten-Simulationen weit verbreitete Open-Source-Software „Repast HPC“ gewählt. Spatio-temporale Ergebnisse werden in einem optimierten MongoDB-Cluster gespeichert und durch unabhängige Analyseskripte ausgewertet. Die verwendete Visualisierung (VAUI) basiert auf der GIS-Bibliothek „SharpMap“ und einigen Standardgraphbibliotheken. Der Anwender muss somit eine limitierte Desktopanwendung ausführen, kann aber die Simulationslaufzeit und -auswertung durch Hinzunahme von weiteren Rechenknoten verringern [LGLW16].

Borgdorff et al. präsentieren eine simulationsgestützte „Urban Movement Analysis“-Plattform, welche wiederrum eine untypisch für Big Data-Anwendungen genutzte PostGIS-Datenbank zur hauptsächlichen Datenspeicherung verwendet. Die Benutzung einer optimierten, verteilten und einfach ersetzbaren PostGIS-Datenbank, wie der Greenplum oder Citius Data, wäre hier denkbar. Zur Datenverarbeitung und -visualisierung stehen dem Benutzer dafür aber eine vorkonfigurierte Weboberfläche mit einem GeoServer und das Jupyter Notebook zur Verfügung, wie auf Abb. 2 zu sehen. Letzteres erlaubt die interaktive Programmierung von beispielweise Python- oder R-Skripten, die auf dem Cluster ausgeführt werden und ein schnelleres Prototyping von Datenanalysen gestattet [BvHD+16]. Sobald ein bestimmtes Datenanalyseskript für alle Benutzer als brauchbar erachtet wird, kann es in den allgemeinen Datenverarbeitungs-Workflow eingebettet werden. Außerdem kann die Allzweck-Datenverarbeitungsplattform Apache Spark als Cluster dynamisch eingebunden werden, um den Großteil der Analyselast abzunehmen [BvHD+16]. Apache Spark erweitert das MapReduce-Paradigma um SQL-Abfragen, die Verarbeitung von Datenströmen, die Berechnung auf Graphen und maschinelles Lernen zu der einheitlichen, sogenannten Berkeley Data Analytics Stack Architektur, welche alle Bibliotheken und hochrangige Komponenten integriert [WBR16]. Viele Post-Processing Aktivitäten bei Simulationen beginnen mit dem Einlesen der Ergebnissets aus Datenbanken. Da aber die Größe dieser Datensets kontinuierlich zunimmt, könnte eine Methodik zur Online-Datenextraktion besonders für cloud-basierte Simulationen, wie sie bei Schützel et al. präsentiert wird, nützlich sein. Die während einer Simulation stattfindende Datenverarbeitung wird genutzt, um den Einsatz langsamer, per-

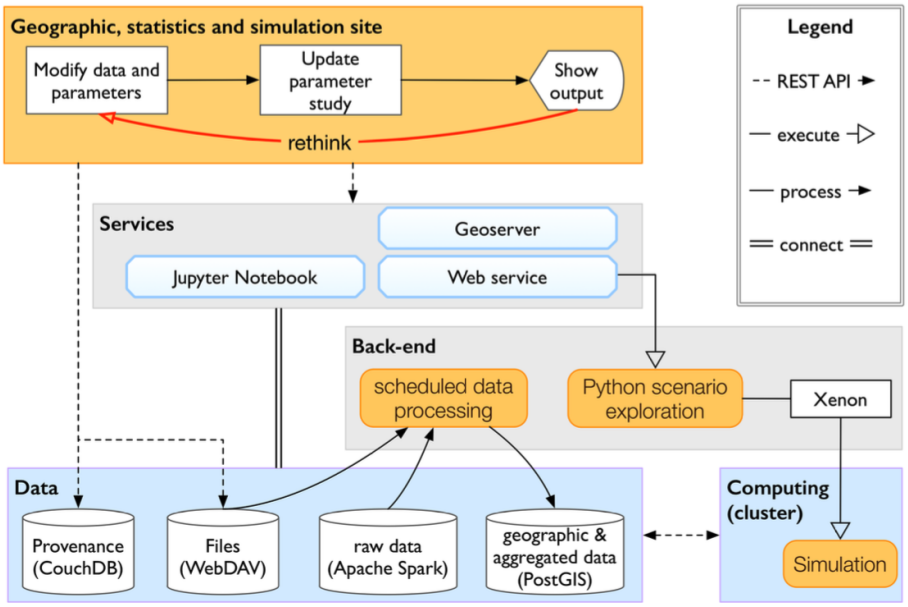


Abb. 2. Architektur der Simulationsplattformen von Borgdoff et al.. Abb. entnommen aus [BvHD+16].

sistenter Speicher zu minimieren [SMU14]. Sind genügend Kapazitäten vorhanden, kann eine komplette Ergebnisausgabe in das für spatio-temporal bekannte HDF-Format oder in ein HDFS-Cluster trotzdem erfolgen. Schützel et al. definieren dafür eine eigenen Processing Graph, ähnlich einer Streamingplattform wie Apache Storm. Weiterhin werden die Simulationsergebnisse von jedem einzelnen Knoten erst an einen sogenannten „Data Collector“ übergeben, welcher ähnlich wie ein Pub/Sub Message Queue-System funktioniert. Die Simulation wird somit weiter entkoppelt und es können externe Datenquellen miteinbezogen werden. Eine praktische Umsetzung eines streamprocessing-gestützten Systems findet sich auch bei Altintas et al., welche Modelle für Lauffeuer simulieren, Echtzeitdaten einbinden und ebenfalls geo-spatiale Visualisierungen besitzen [ABDC+15].

3.1 Geospatiale und temporale Repäsentation

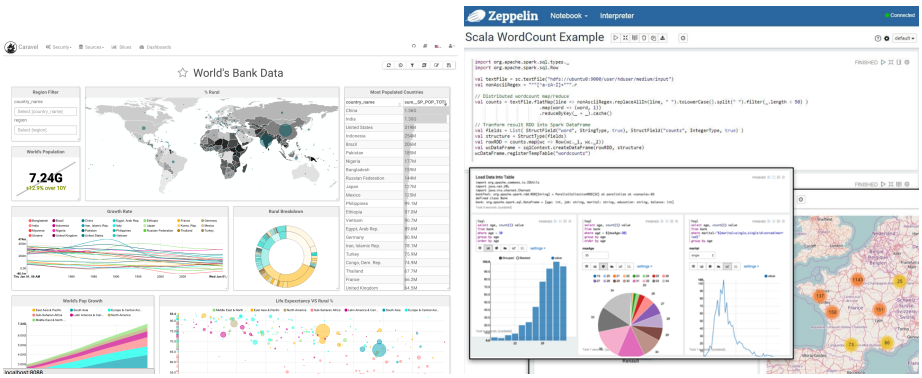
Abseits von Simulationsplattformen gibt es einige cloud-basierte, geospatiale Plattformen wie KAVE oder TrackLab, um Daten zu ingestieren, zu analysieren und zu visualiseren [ABDC+15]. Zugrundeliegende Key-value Stores (KVS) versprechen hier Fortschritte, bieten sie doch Skalierbarkeit, Fehlertoleranz, Verfügbarkeit und hohe zufällige Lese-/Schreibvorgänge [VL16]. Das KVS Design orientiert sich wie bei HBase, Cassandra und Accumulo an Googles BigTable. Bei

dieser Systemarchitektur werden Daten auf der Ebene von Dateneinträgen bearbeitet. Jeder Dateneintrag erhält dazu einen Schlüssel sowie mehrere Attribute bzw. Werte. Schnelle Zugriffe auf einzelne Dateneinträge sind mit ihnen ebenfalls möglich, da alle Einträge nach Schlüsseln sortiert vorliegen [EM+16].

Sie bieten indessen keine native Unterstützung für geospatiale und temporale Abfragen. Einige vorgeschlagene Systeme wie SpatialHaddop und Hadoop-GIS basieren auf MapReduce und arbeiten lediglich auf flüchtigen In-Memory Indizes. Persistente Indizes, die bspw. auf K-d-Tree, Quad- oder R-Tree und Geohash Index-Verfahren basieren, wurden durch Erweiterungen in die KVS eingebunden, um das Einfügen und Löschen in nahezu Echtzeit zu gestatten [EM+16]. Experimentelle Systeme wie STEHIX und MD-HBase implementieren etwa einen multidimensionalen Index mit Hilfe von Z-Sortierung und multidimensionale Indexstrukturen in KVS [VL16]. Sie unterstützen aber keine evolutionsbezogenen, räumlichen Abfragen, bei denen sich räumliche Beziehungen im Zeitverlauf verändern. Bei GeoMesa wird eine bestimmte, Schlüssel-Wert-Paaren zugrundeliegende Ordnung so genutzt, das räumliche Einträge mit einer raumfüllenden Kurve wie der Z-Kurve linear angeordnet werden [HZEF16]. Der linearisierte Wert wird dabei für einen Teil des Schlüssels des Indexes verwendet. Dies stellt sicher, dass räumlich nahe beieinanderliegende Schlüssel auch nahe beieinander oder nach einer anderen verteilten Ordnung auf die jeweilige Datenbankpartition geschrieben werden [HAE+15]. Aufbauend auf diesen Indizes können punktbasierte, rangbasierte und kNN-Abfragen auf effiziente Weise implementiert werden [EM15].

3.2 Generische Datenvisualisierungssysteme

Visuellen Zugang zu den beschriebenen Datenquellen bieten beispielweise kommerzielle Buisness Intelligene Analyseprogramme, wie SAS Visual Analytics, Pentaho oder Tableau, an. Da diese zu meist proprietär und sich schlecht in einer eigenen Cloudumgebung ausrollen lassen, werden hier zwei Kategorien von quelloffenen Systemen mit Dashboardfähigkeiten vorgestellt. Einerseits sind es Abfrageeditoren, die Datenbankkonnektoren besitzen und den Zugang zur DB mit Abfragesprachen wie SQL abstrahieren. So können dynamische Dashboards wie beim Beispiel von Superset, s.Abb. 3a erstellt werden. Metabase und Kibana/Grafana erleichtern die Bedienung noch weiter, indem der Benutzer aus generischen, sich dem Datenbestand anpassten Feldern, seine Abfragen zusammenklicken kann. Andererseits haben sich programmierbare Online-Editoren bzw. Notebooks wie das bereits erwähnte Jupyter bewährt, die jegliche Berechnungen auf einem Cluster ausführen und nur das Ergebnis mit entsprechenden Visualisierungen auf der Benutzeroberfläche anzeigen. Apache HUE, speziell für das Hadoop-Ecosystem gebaut, zählt eher zu den typischen SQL-Editoren, entwickelt sich zuletzt aber mit einer Scala, Java und Python-Integration zu einem weiteren Bewerber für Notebooks. Das in Abb. 3b zu sehende Apache Zeppelin bietet aufgrund der moderneren Webtechnologien und nativen Unterstützung für



(a) Airbnb Superset. Abb. entnommen aus [sup].

(b) Apache Zeppelin. Abb. entnommen aus [zepa] und [zepb]

Abb. 3. Beispielumsetzungen eines Dashboards und Notebooks.

Spark sowie Datenbankinterpreter für Cassandra oder HBase eine entwicklungsfreundliche Umsetzung. Für interdisziplinär und kollaborativ arbeitende Wissenschaftler ist diese Art von webbasierter Oberfläche für gemeinsame Analysen der gleichen Daten eine akzeptierte Lösung [BvHD+16]. Sie finden sich aber, wie gezeigt, sehr selten und schlecht integriert in MAS-Umgebungen wieder.

4 Konzept und Ausführung der angestrebten Plattform

Die verteilte Architektur von MARS mit dem cloud-basierten Servicemodell ist auch hier wiederzufinden. Alle Komponenten der Plattform sind mithilfe des Container-Verwaltungstools Kubernetes umgesetzt, welches für die Ausführung, Wartung und die Skalierung der verpackten Anwendungen genutzt wird.

Der präsentierte Ansatz für die Big Data Datenverarbeitung in beinahe Echtzeit versucht das Hauptproblem der Analyse großer, kontinuierlicher Datensätze zu lösen. Die Lambda Architektur ist dafür eine der neuesten und jüngst auch besonders populären Methoden [TR14]. Drei klar zu unterscheidende Ebenen für Stapel- und Echtzeitverarbeitung sowie Abfragedienste garantieren eine saubere Trennung der einzelnen Komponenten und Funktionen. Für jede dieser Ebenen steht eine große Auswahl an implementierbaren und etablierten Werkzeugen bereit. Viele dieser Lösungen sind bereits seit Jahren auf dem Markt und für ihre Zuverlässigkeit bekannt [TR14].

Noch vor der Simulationsausführung werden programmatisch die Plattformkomponenten wie beispielsweise Kafka und Spark für einen neuen Simulationslauf konfiguriert und die Datenbanktabellen populiert. Der Benutzer kann auf der

Weboberfläche Einstellungen bezüglich der von ihm erwarteten Simulationsausgabe treffen. Die Oberfläche ist in die einheitlichen MARS Weboberfläche integriert, um einen Kontext- bzw. Zugehörigkeitswechsel zu vermeiden. Da für die benutzten Simulationsmodelle alle Rohdaten pro festgelegtem Zeitschritt und Agenten herausgeschrieben werden und die Simulationsknoten möglichst schnell entlastet werden sollen, schreibt jede Simulationseinheit im Binärformat die Daten, wie in Abb. 4 zu sehen ist, in eine Apache Kafka Queue.

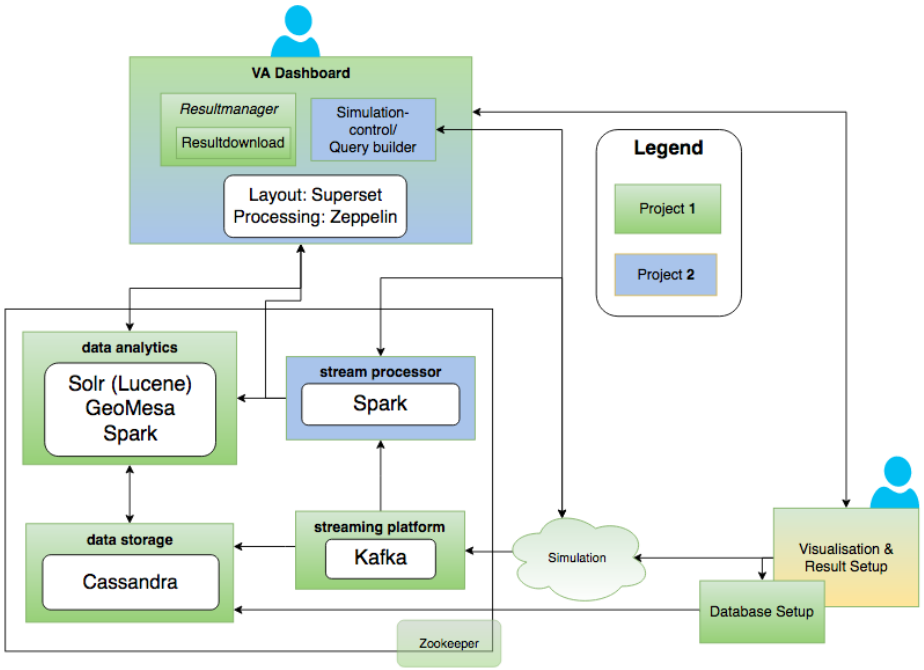


Abb. 4. Ganzheitliche Architektur der angestrebten Visual Analytics-Plattform.

Von dort aus wird durch registrierte, sogenannte Kafka Consumer der Cassandra Datenbankcluster befüllt sowie die Daten an den jeweiligen Stream Processor übergeben. Derzeit wird Spark Streaming eingesetzt, um eine einheitliche Programmierschnittstelle bieten zu können. Wiederholt auszuführende Aggregationen können so beispielsweise programmatisch in das System eingepflegt werden. Der abstrahierte Datenbankzugriff über das generische Dashboard erfolgt über SparkSQL für nicht geospatiale Abfragen, CQL mit dem Lucene Indexer oder über GeoMesa für geospatial und temporal indexierte Daten.

Erste Ergebnisse einer Simulation können darüber hinaus in einem Zeppelin Notebook und dem Prototypen, welcher in Kap. 2 vorgestellt wurde, begutachtet werden. Die Architektur ist weitläufig mit der Simulation verknüpft und bedarf

einiger Optimierungen in Hinsicht auf die internen Umstellungen der Simulationsengine in MARS. Die generische Bereitstellung der Datenumgebung an die Notebooks und Dashboardelemente stehen aus, so wie auch die Integration und Überführung der Visualisierungen vom Prototypen und weiteren bisher noch nicht unterstützten Visualisierungen mit entsprechendem Logikcode. Eine programmatische Einbindung und Echtzeitauswertung der Simulation mit Hilfe des Stream-Processors folgt.

5 Evaluierung und Testumgebung

Die Ausfallsicherheit und Zuverlässigkeit der abgekoppelten Komponenten, die jeweiligen API-Endpunkte und die Benutzbarkeit sowie der Nutzen der Weboberfläche müssen überprüft, gemessen und bewertet werden. Ein Problem der komponentenbasierten Architektur ist die deutlich erschwerte Konfiguration für einen Simulationsdurchlauf und dessen Auswertung. Die Umsetzung mit Docker-Containern und Microservices erwies sich als förderlich. Durch eigene Testprogramme für jede Hauptkomponente können die Integrität gewährleistet und Performancemerkmale erörtert werden. So konnte ein höherer Schreibdurchsatz pro Simulation bei gleichbleibenden Ressourcen erlangt werden, welcher dazu noch im Gegensatz zur vorherigen Lösung linear zur Knotenanzahl skaliert. Weitere Merkmale und Ergebnisse in Hinsicht auf die Performance und erste Erfahrungen mit der Dashboard-Oberfläche finden sich in den Projektberichten wieder. Ein komplexes, ökologisches Simulationsmodell dient dazu, die Fähigkeiten der VA-Anwendung zu demonstrieren. Im Verlauf der Projekte stellte sich heraus, dass technische Anwender einen direkten Zugang zu den Simulations- und Datenanalyse-Infrastrukturen benötigen. Die Integration von Notebooks erwies sich hier als sehr nützlich. Um das volle Spektrum der Plattform nutzen zu können, brauchen Modellierer bisher außerdem Unterstützung durch Dritte.

6 Chancen und Thesis Outline

Beinahe alle bisher entwickelten, auf Cloud Computing-basierenden Simulationssysteme stellen eine Vielfalt an Funktionsmöglichkeiten zum Erstellen und Verwalten von Modellen sowie der verteilten Ausführung zur Verfügung. Die primären Forschungsziele wurden meist auf Performancevergleiche und spezifische Domänen ausgerichtet, selten aber wird die Interaktion oder die explorative Visualisierung generischer Modelle mit geospatialen Bezug in Betracht gezogen. Mit der hier vorgestellten Plattform erreicht man durch effektives Datenmanagement verkürzte Simulationslaufzeiten und nahe Echtzeit Abfragen auf diese Daten. Durch die vorgestellten Visualisierungs- und Auswertungskomponenten könnten verschiedene Benutzer unabhängig ihrer technischen Erfahrungen und

Interessen mit den Simulationsergebnissen arbeiten. Interaktive Visualisierungen können in Zukunft sehr leicht mit Machine Learning Algorithmen kombiniert werden. Einige hier vorgeschlagene Implementationen könnten bei Erfolg der Open-Source-Gemeinschaft zu Gute kommen. Neben der Anwendung im Kontext von MAS, ließe sich die Plattform mit wenigen Modifikationen auch interdisziplinär im Bereich der Internet of Things betreiben. Statt Agenten hätte man hier beispielsweise Sensoren.

Da das Anwendungsspektrum groß ist und die Plattform viele Ansatzpunkte für tiefergehende Forschungsarbeiten bietet, soll für die Masterarbeit erst eine Themeneingrenzung in Bezug auf Multi-Agenten-Simulationen stattfinden. Es muss erörtert werden, welche spezielle Ausrichtung beispielsweise ein Dashboard haben muss, damit es dem Simulationsmodellierer bei der Entscheidungsfindung hilft. Meinen Fokus möchte ich weiter auf die Visual Analytics legen und Echtzeit Visualisierungen in Kombination mit Ad-hoc Analysen bereitstellen. Zu klärende Fragen finden sich dabei einerseits in weiteren Datenreduktionsverfahren, Level of Detail-Visualisierungen und spatio-temporale Movement Analysis [DBC+15]. Dies könnte im Endeffekt zu einer Ausprägung der Simulationskontrolle und -validierung führen.

7 Risiken

Obwohl die in Kap. 4 präsentierte Plattform durch die Lambda Architektur klar strukturiert ist, erfordert sie eine Orchestrierung und ist mit einigem Integrationsaufwand verbunden. Die Interaktion zwischen den einzelnen Ebenen und der restlichen MARS-Struktur muss richtig umgesetzt werden. Darüber hinaus erfordert auch die Interaktion bestimmter Elemente innerhalb einer Ebene den Einsatz verschiedener Protokolle und Kommunikationstechniken. In Big Data Umgebungen bedeutet dies, dass man bei der Integration von verteilten Datenverarbeitungssystemen sowohl Skalierbarkeit als auch einen zuverlässigen Betrieb sicherstellen muss. Wird die Plattform in externen Cloudumgebungen oder von externen Benutzern betrieben, stellt sich hier schnell die Frage, ob die Kosten für alle Rechenknoten den Nutzen dieser breitflächigen Plattform übersteigen. Es ist schwer zu sagen, wann die Plattform für gut befunden wird, da die Bewertungskriterien und die Bedeutung einer Multi-Agenten-Simulation für ein Dashboard noch nicht klar definiert wurden. Dies ist besonders für die Weboberfläche wichtig, um eine klare Abgrenzung und eine andere Erwartungshaltung gegenüber gängigen Business Intelligence Tools zu schaffen. Ob der generische Ansatz, welcher in vielen Komponenten benutzt wurde, mehr von der Community als ein gezielter, für eine bestimmte Domäne entwickelter Ansatz akzeptiert wird, ist noch zu klären. Zur Risikoverminderung kann hierbei eine enge Zusammenarbeit mit Modellierern, Fachexperten und den Entwicklern der Open-Source-Lösungen dienen. Es werden bereits Anforderungen und Proposals mit Partnern des eScience Center und der University of Florida diskutiert.

8 Fazit

In dieser Arbeit wird eine Plattform präsentiert, die Komponenten aus der Visual Analytics, geo-spatiale und temporale Visualisierungen sowie large-scale Multi-Agenten-Simulationen zusammenführt. Kern und Fokus dieser Plattform ist die effiziente Datenspeicherung, -verwaltung und -auswertung. Durch die gesammelten Erfahrungen bei der Umsetzung des Prototypen und den Vergleichen zu anderen Forschungsarbeiten wurden bestimmte Open-Source-Komponenten bewertet und ausgewählt. Kollaboratives Arbeiten von verschiedenen Modellierern wird beispielsweise durch den Einsatz von generischen Dashboards gefördert. In zukünftigen Arbeiten sollen für diese Plattform die Echtzeit-Datensammlung und interaktive Datenanalyse im Kontext der Visual Analytics thematisch vertieft werden, da diese als Schlüsselargumente für eine effektive Simulationsauswertung herausgearbeitet wurden.

Literatur

[AA13] Natalia Andrienko and Gennady Andrienko. Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, 12(1):3–24, January 2013.

[ABDC+15] Ilkay Altintas, Jessica Block, Raymond De Callafon, Daniel Crawl, Charles Cowart, Amarnath Gupta, Mai Nguyen, Hans-Werner Braun, Jurgen Schulze, Michael Gollner, et al. Towards an integrated cyberinfrastructure for scalable data-driven monitoring, dynamic prediction and resilience of wildfires. *Procedia Computer Science*, 51:1633–1642, 2015.

[BAA+16] Andrew C Bauer, Hasan Abbasi, James Ahrens, Hank Childs, Berk Geveci, Scott Klasky, Kenneth Moreland, Patrick O'Leary, Venkatram Vishwanath, Brad Whitlock, et al. In situ methods, infrastructures, and applications on high performance computing platforms. In *Computer Graphics Forum*, volume 35, pages 577–597. Wiley Online Library, 2016.

[BvHD+16] J Borgdorff, W van Hage, LJ Dijkstra, E Mancini, and MH Lees. Simulation-supported urban movement analysis. 2016.

[CAD+14] Lauren N Carroll, Alan P Au, Landon Todd Detwiler, Tsung-chieh Fu, Ian S Painter, and Neil F Abernethy. Visualization and analytics tools for infectious disease epidemiology: a systematic review. *Journal of biomedical informatics*, 51:287–298, 2014.

[DBC+15] Urška Demšar, Kevin Buchin, Francesca Cagnacci, Kamran Safi, Bettina Speckmann, Nico Van de Weghe, Daniel Weiskopf, and Robert Weibel. Analysis and visualisation of movement: an interdisciplinary review. *Movement ecology*, 3(1):5, 2015.

[Dyb16] Janus Dybulla. Visual analytics zur unterstuetzung von multi-agenten simulationen, 2016.

[EM15] Ahmed Eldawy and Mohamed F Mokbel. The era of big spatial data. In *Data Engineering Workshops (ICDEW), 2015 31st IEEE International Conference on*, pages 42–49. IEEE, 2015.

[EM+16] Ahmed Eldawy, Mohamed F Mokbel, et al. The era of big spatial data: A survey. *Foundations and Trends® in Databases*, 6(3-4):163–273, 2016.

[GSL+11] László Gulyás12, Attila Szabó, Richárd Legéndi, Tamás Máhr, Rajmund Bocsi, and George Kampis. Tools for large scale (distributed) agent-based computational experiments. 2011.

[HAE+15] James N Hughes, Andrew Annex, Christopher N Eichelberger, Anthony Fox, Andrew Hulbert, and Michael Ronquest. Geomesa: a distributed architecture for spatio-temporal fusion. In *SPIE Defense+ Security*, pages 94730F–94730F. International Society for Optics and Photonics, 2015.

[HATCea16] Christian Huening, Mitja Adebahr, Thomas Thiel-Clemen, and Janus Dybulla et al. Modeling and simulation as a service with the massive multi-agent system mars. To appear in Proceedings of the 2016 Spring Simulation Multiconference, 2016.

[HZEF16] James N Hughes, Matthew D Zimmerman, Christopher N Eichelberger, and Anthony D Fox. A survey of techniques and open-source tools for processing streams of spatio-temporal events. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, page 6. ACM, 2016.

[LDC+16] Songnian Li, Suzana Dragicevic, Francesc Antón Castro, Monika Sester, Stephan Winter, Arzu Coltekin, Christopher Pettit, Bin Jiang, James Haworth, Alfred Stein, et al. Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:119–133, 2016.

[LGLW16] Zhenqiang Li, Xuefeng Guan, Rui Li, and Huayi Wu. 4d-sas: A distributed dynamic-data driven simulation and analysis system for massive spatial agent-based modeling. *ISPRS International Journal of Geo-Information*, 5(4):42, 2016.

[LJH13] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430. Wiley Online Library, 2013.

[PBH+15] Alexandre Perrot, Romain Bourqui, Nicolas Hanusse, Frédéric Lalanne, and David Auber. Large Interactive Visualization of Density Functions on Big Data Infrastructure. In *5th IEEE Symposium on Large Data Analysis and Visualization*, Chicago, United States, October 2015. IEEE.

[RHWU14] Stefan Rybacki, Fiete Haack, Karsten Wolf, and Adelinde M. Uhrmacher. Developing simulation models - from conceptual to executable model and back - an artifact-based workflow approach. In *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*, SIMUTools '14, pages 21–30, ICST, Brussels, Belgium, Belgium, 2014. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[SMU14] Johannes Schützel, Holger Meyer, and Adelinde M. Uhrmacher. A stream-based architecture for the management and on-line analysis of unbounded amounts of simulation data. In *Proceedings of the 2Nd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM PADS '14, pages 83–94, New York, NY, USA, 2014. ACM.

[sup] Superset dashboard. `http://imgur.com/SAhDJCI` - zuletzt aufgerufen am 10.01.2017.

[TR14] Bartlomiej Twardowski and Dominik Ryzko. Multi-agent architecture for real-time big data processing. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 3, pages 333–337. IEEE, 2014.

[VL16] Hong Van Le. Distributed moving objects database based on key–value stores. 2016.

[WBR16] Caesar Wu, Rajkumar Buyya, and Kotagiri Ramamohanarao. Big data analytics= machine learning+ cloud computing. *arXiv preprint arXiv:1601.03115*, 2016.

[YG16] Can Yang and Gyozo Gidófalvi. Interactive visual exploration of most likely movements. In *the 19th AGILE International Conference on Geographic Information Science*, 2016.

[YYH+17] Chaowei Yang, Manzhu Yu, Fei Hu, Yongyao Jiang, and Yun Li. Utilizing cloud computing to address big geospatial data challenges. *Computers, Environment and Urban Systems*, 61:120–128, 2017.

[zepa] Zeppelin notebook - 1. `https://aws.amazon.com/blogs/big-data/building-a-recommendation-engine-with-spark-ml-on-amazon-emr-\-using-zeppelin/` - zuletzt aufgerufen am 10.01.2017.

[zepb] Zeppelin notebook - 2. `https://www.zeppelinhub.com/images/b89432b04a\-74bc2ba765fab33db9a249.png` - zuletzt aufgerufen am 10.01.2017.

[ZVCK16] Daniel Zehe, Vaisagh Viswanathan, Wentong Cai, and Alois Knoll. Online data extraction for large-scale agent-based simulations. In *Proceedings of the 2016 Annual ACM Conference on SIGSIM Principles of Advanced Discrete Simulation*, SIGSIM-PADS '16, pages 69–78, New York, NY, USA, 2016. ACM.