

Bridging the Gap between Serving and Analytics in Scalable Web Applications

Panagiotis Garefalakis

Imperial College London
pg1712@imperial.ac.uk

Raul Castro Fernandez

Imperial College London
rc3011@doc.ic.ac.uk

Peter Pietzuch

Imperial College London
prp@doc.ic.ac.uk

Abstract. Web applications that include personalised recommendations, targeted advertising, and other analytics functions must maintain complex prediction models, which are trained over large datasets. Such applications typically separate stored data into *offline* and *online* data based on its time to compute and freshness requirements. To serve requests robustly and with low latency, applications then cache data from the analytics layer, constructing responses from this data; to train models and offer analytics, they use asynchronous offline computation in the analytics layer, which leads to stale data being served to clients.

Instead, our goal is to offer a unified model to developers when writing web applications that serve data while using big data analytics. Our idea is to express the online and offline logic of a web application as a single *stateful distributed dataflow graph*. The state of the dataflow computation is then expressed as *In-memory Web Objects* (IWOs), which are directly accessible as persistent objects by the application. This means that the application can exploit data-parallel processing for compute-intensive requests, e.g. when training complex models, while serving results with low latency from IWOs.

Motivation. Modern web applications must offer low-latency responses, which typically means that they pre-compute computationally-expensive *analytics* tasks such as personalised recommendations using asynchronous back-end systems. These tasks are decoupled from the critical path of *serving* web requests, and, for serving, developers must load pre-computed results into scalable stores such as distributed key/value stores. The pre-computation is performed by data-parallel frameworks such as Hadoop or Spark.

Challenges. Despite achieving low latency, the above approach has a number of limitations: (i) decoupling data analytics from serving means that results can be stale, which has a negative impact on time-critical data such as advertisements. More than 70% of all Hadoop jobs running at LinkedIn [3] use key-value stores as egress mechanism; (ii) caching data, while reducing the read load, requires the construction of complex queries, involving multiple back-end stores; and (iii) a variety of stores and back-end systems must be managed and scaled out independently, which has proven to be hard, error-prone and inefficient.

Approach. Our goal is to provide a unified model for web applications by exposing a common object-based interface to the developers for handling offline and online data. A web application framework, such as Play for Java, can directly manipulate persistent objects, which we extend to become *In-memory Web Objects* (IWOs). IWOs are computed in data-parallel fashion, either on-demand when a web request is handled, or asynchronously when representing previously computed, cached data. They are implemented as in-memory state in a stateful distributed dataflow model [1]. Since IWOs are maintained in-memory but manipulated in a data-parallel fashion, they can satisfy the requirements of offline and online data processing in web applications.

We describe an initial design of a framework that supports stateful dataflow graphs with IWOs based on SEEP, an open-source, data-parallel processing platform. We show that, using a source-to-source compiler [2], it is possible to automatically synthesise the dataflow graphs for existing Java-based web applications, thus benefiting from data-parallelism for serving computationally-expensive requests while maintaining analytics over large amounts of data.

References

- [1] R. Castro Fernandez, M. Migliavacca, E. Kalyvianaki, and P. Pietzuch. Integrating Scale Out and Fault Tolerance in Stream Processing using Operator State Management. In *ACM SIGMOD*, 2013.
- [2] R. Castro Fernandez, M. Migliavacca, E. Kalyvianaki, and P. Pietzuch. Making State Explicit for Imperative Big Data Processing. In *USENIX ATC*, 2014.
- [3] R. Sumbaly, J. Kreps, and S. Shah. The “Big Data” Ecosystem at LinkedIn. In *ACM SIGMOD*, 2013.

Bridging the Gap between Serving and Analytics in Scalable Web Applications

Panagiotis Garefalakis, Raul Castro Fernandez, Peter Pietzuch

Large-Scale Distributed Systems (LSDS) Group, Department of Computing

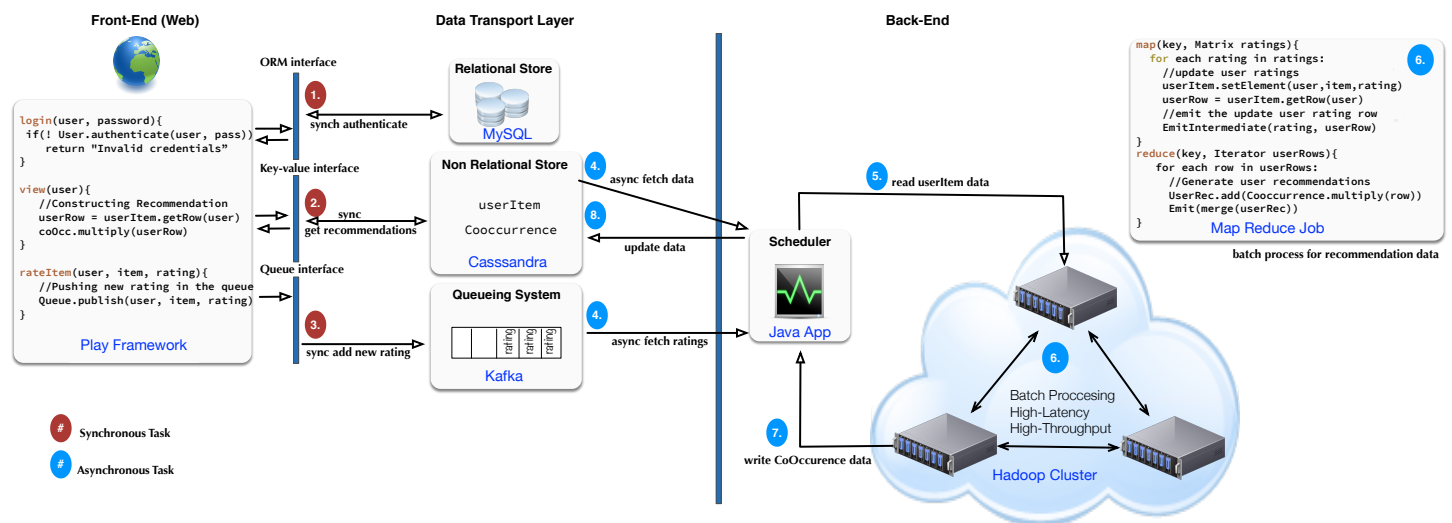
Imperial College London

pg1712@imperial.ac.uk, rc3011@doc.ic.ac.uk, prp@doc.ic.ac.uk

Motivation

- Most modern web and mobile applications today offer highly personalised services generating large amounts of data
- Stored data is separated into offline and online data, based on its generation cost and freshness requirements
- Data resides on different storage layers and is processed by different systems to hide the underlying complexity
- Serving** and **Analytics** layers are decoupled in order to serve requests with the minimum processing overhead

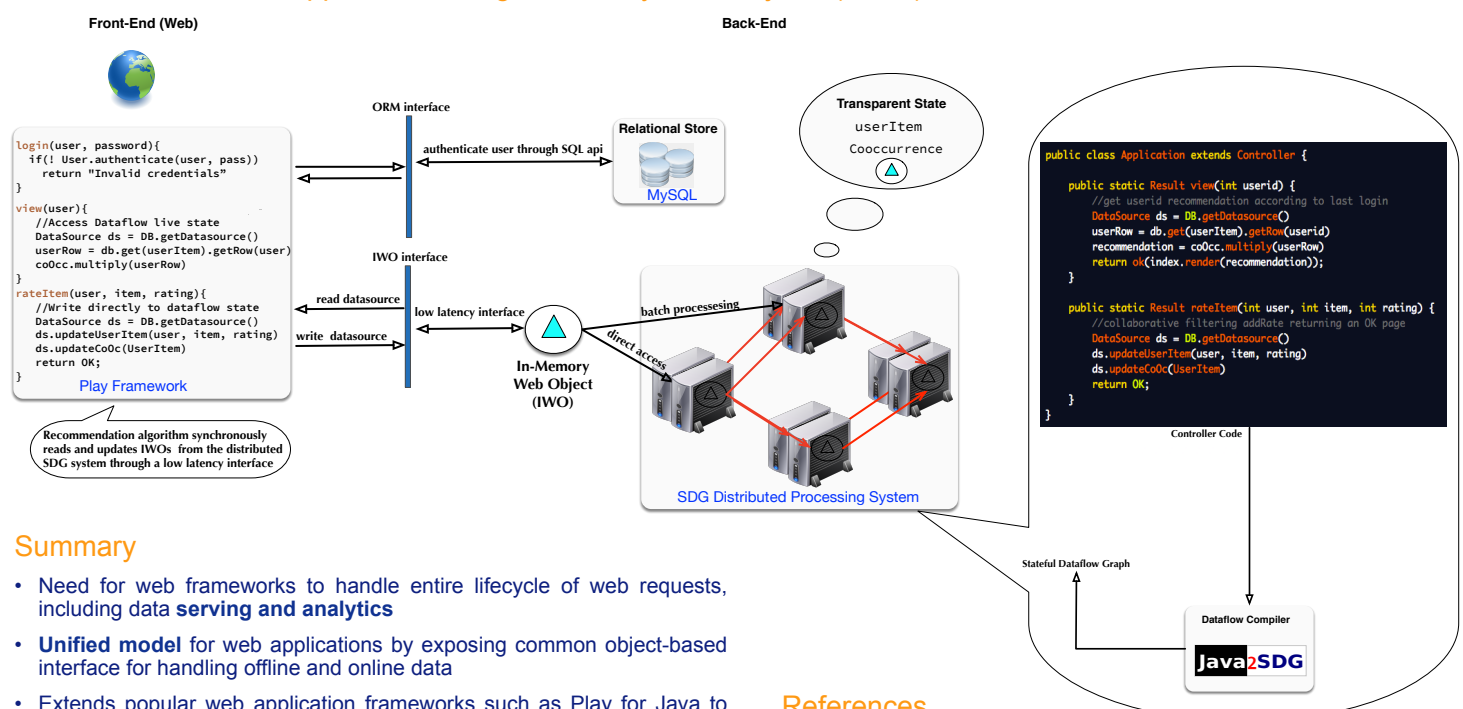
Typical Web Application Today



Challenges

- Serving stale data can have negative impact on **time critical** tasks
- Building web responses using multiple systems involves joining information from **complex** back-end systems
- Increased developers' **effort** to learn and program these systems
- High complexity to maintain, monitor and scale a **variety of systems** independently

Dataflow-Based Web Application using In-memory Web Objects (IWOs)



Summary

- Need for web frameworks to handle entire lifecycle of web requests, including data **serving and analytics**
- Unified model** for web applications by exposing common object-based interface for handling offline and online data
- Extends popular web application frameworks such as Play for Java to directly manipulate **In-Memory Web Objects (IWOs)** instead of plain persistent objects

Acknowledgments

This work was partially supported by a PhD CASE Award funded by EPSRC/BAE Systems and by the High-Performance and Embedded Distributed Systems Centre for Doctoral Training (CDT) HiPEDS programme.

References

- R. Sumbaly, J. Kreps, and S. Shah. The "Big Data" Ecosystem at LinkedIn. In SIGMOD, 2013.
- R. C. Fernandez, M. Migliavacca, et al. Integrating Scale Out and Fault Tolerance in Stream Processing using Operator State Management. In SIGMOD, 2013.
- R. C. Fernandez, M. Migliavacca, E. Kalyvanaki, and P. Pietzuch. Making state explicit for imperative big data processing. In USENIX ATC, 2014.