Effizienz-Optimierung
daten-intensiver Data
Mashups am Beispiel
von Map-Reduce

Pascal Hirmer
Towards optimizing the efficiency of data-intensive data mashups based on the example of Map-Reduce

Pascal Hirmer
Motivation

Big Data

• Big Data: volume and complexity of data highly increases
  • New paradigms: Internet of Things, Industrie 4.0, Data Lakes, ...

• It is important to gain knowledge through data processing and analysis (knowledge discovery)

• But: gaining knowledge is difficult because of the (at least) **three Vs of Big Data**:
  • Volume
  • Variety
  • Velocity
Data Mashups - Definition

- Goal: flow-based processing, analytics, and integration of data
- Modeling of data operations based on Pipes and Filters
- Famous example: Yahoo! Pipes
Motivation
Data Processing Tools

• Data Mashup tools, ETL tools, and data analytics tools (e.g. KNIME) offer means to process and analyze data
• Focus on approaches that support abstract modeling based on the pipes and filters pattern
  • nodes: data operations (e.g., extraction, transformation, analysis)
  • edges: data flow
  • nodes are associated with services that process the data (orchestrated by workflows)
• Offer an explorative means to process data

• Focus lies on the Open Source Data Mashup Tool FlexMash developed at the Uni Stuttgart
  • Concepts are also applicable to different approaches for data processing
Motivation

• **Overall goal of this work:** Increasing the efficiency of service-based data processing

• **State of the art:** data processing "in-service" (memory) $\rightarrow$ scalability / memory issues

• **Approach in a nutshell:**
  • Move data processing on computing clusters and process data in parallel
  • Integration of modern data processing techniques and technologies (Map-Reduce, Apache Spark, …)
  • Coping with the generated overhead (where is the cost-value limit?)
FlexMash

Cloud-based execution

Mashup Execution Environments
- Robust
- Time-Critical
- Secure
- Robust & Secure
- ...

Mashup Result

Domain-specific Modeling
Pattern Selection
Pattern-based Transformation and Execution
Visualization

Mashup Modeler
FlexMash Modeling Tool
Mashup Plan
Pattern Selection & Combination
FlexMash – Graphical User Interface

Download FlexMash on Github: https://github.com/hirm erpl/FlexMash
Main contribution (I)

Mashup Plan (non-executable)

Executable representation of the data flow model

Service runtime

Parallel data processing based on computing clusters

Parallel data processing

in-service

extract filter join analyze
Main contribution – decision: in-service vs. distributed/parallel
Conclusion and future work

• First approach to increase the efficiency of service-based data processing tools
• Large efficiency advantages enabled through parallelization
• Finding the cost-value limit is difficult

• Future/ongoing work
  • Conducting measurements for comparison and finding cost-value limit
  • Concretizing the concepts
  • Generation of Map-Reduce jobs
Questions & Discussion
Thank you!

Pascal Hirmer

E-Mail Pascal.Hirmer@ipvs.uni-stuttgart.de
Telefon +49 (0) 711 685-88297
Fax +49 (0) 711 685-78217

Universität Stuttgart
Pascal.Hirmer@ipvs.uni-stuttgart.de
Universitätsstraße 38, 70569 Stuttgart, Germany