

Master Thesis Proposal

Implementation of a user interface to facilitate the creation and execution of PACTs (Draft title)

Mathias Nitzsche

[nitzsche@informatik.hu-berlin.de]

Humboldt University Berlin
Department of Computer Science
Databases and Information Systems

Supervisors: Mathias Peters, Rico Bergmann, Prof. Johann-Christoph Freytag

Introduction

The "Stratosphere" Collaborative is a joint research initiative of the Technische Universität Berlin, the Humboldt Universität zu Berlin, and the Hasso-Plattner-Institut in Potsdam. In the project the involved scientists strive to develop a novel, database-inspired approach to analyze, aggregate, and query very large collections of either textual or (semi-)structured data. Superior to existing parallel databases the final system will allow schema free processing and furthermore fully utilize the power of virtualized, massively parallel cluster architectures in a fault tolerant and transparent fashion. [5]

Existing approaches in this direction, among them Google's Map/Reduce [2] and its open source counterpart Hadoop, do not offer the expressive power needed for complex database-like operations (eg Joins). For this reason a new programming model called "Parallelization Contracts" (PACTs) and the scalable execution engine Nephele have been developed.

A PACT consists of one second-order function called input contract and an optional output contract. An input contract takes a first order function with user code and one or more data sets as parameters. Therefore input contracts generalize the well-known Map/Reduce programming model with further functions (beside Map and Reduce, Cross, Match and others exist) and extend it with the possibility to pass more than one data set. The optional output contract allows developers to guarantee certain behaviour of the user code regarding the properties of the output data. [1]

In order to execute a PACT program it has to be submitted to the PACT compiler, which translates it into a data flow program, modelled as directed acyclic graph (DAG). Within this DAG vertices

represent subtasks and edges represent communication channels between these subtasks. Each subtask is a sequential program that reads data from its input channel and writes data to its output channel. At that stage, cost-based optimizations and the additional semantic information from the output contract, like “Same-Key” or “Unique-Key” are exploited, to generate more efficient data flow graphs. [4]

The Nephele System is then used to execute the data flow graphs on a distributed infrastructure. Since the initial DAG does not reflect parallel execution, Nephele generates the parallel data flow graph by spanning the received DAG to the desired degree of parallelism. During execution Nephele takes care of resource scheduling, task distribution, communication and synchronisation. Input- and Output-data is stored in the distributed HaDoo FileSystem. [3]

Motivation

The described system was already presented at different conferences, although it is still under development. While the implementation of several components has not been started, others are working fine but are complicated to use and require deep system knowledge. One example is the process of creating and executing a PACT, by performing the following steps:

- Setup the development environment:
 - download and install the Eclipse IDE
 - install and configure maven and git plugins
 - fetch the current stratosphere repository
- Create the PACT:
 - create a new java class, which extends two base classes
 - implement needed methods and the custom code
 - use correct annotations for output contracts
 - define execution plan
- Compile and execute:
 - compile the class and its dependencies into a java archive (.jar)
 - upload this jar using the Nephele user interface and execute it

Obviously this process is time consuming, hard to use, fault-prone and most important in its current state only available to a limited group of people.

Goal

The goal of this master thesis is to implement a solution which enables users to create, maintain, share and execute PACTs, without the need of installing or configuring a local development environment. Based on the identified shortcoming of the PACT creation process a new graphical user interface will be implemented, with the following basic requirements:

- The solution needs to be browser based (thus of course cross browser and cross-operating system compatible)

- A secure and user specific login has to be provided.
- The modelling environment for PACTs must be easy to use and allow the creation of all entities, like tasks with custom code, channel types and output contracts.
- A persistence functionality must be provided to load and save PACTs.

Not within the scope are the following functions:

- Collaborative/simultaneous multi user editing of PACTs.
- Support of mobile devices.
- A multiple language interface.

The software creation process will be carried out in an engineering-style manner, going through the different phases of the development process and using existing software development, testing and documentation tools like Eclipse, Maven, Hudson, Git, Trac and Wiki.

The outcome of the project will be the following deliverables:

- an academic paper (master thesis),
- a fully implemented and tested working software, including test cases,
- and a software documentation (within the code and as far as needed as a dedicated document)

Timeline

Currently it is planned to conduct the described work within the next 6 months - thus until October 2011. As soon as possible a detailed time line including mile stones will be created. During this time M. Nitzsche will be advised on a regular basis by M. Peters and R. Bergmann. Prof. J. Freytag will be the academic supervisor. A second academic supervisor will be named as soon as possible.

References

[1] MapReduce and PACT - Comparing Data Parallel Programming Models

Alexander Alexandrov, Stephan Ewen, Max Heimes, Fabian Hueske, Odej Kao, Volker Markl, Erik Nijkamp, Daniel Warneke

In Proceedings of Datenbanksysteme für Business, Technologie und Web (BTW) 2011, 25-44

[2] MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

In OSDI, pages 137–150, 2004.

[3] Massively Parallel Data Analysis with PACTs on Nephele

Alexander Alexandrov, Dominic Battré, Stephan Ewen, Max Heimes, Fabian Hueske, Odej Kao, Volker Markl, Erik Nijkamp, Daniel Warneke

PVLDB Vol. 3, No. 2, 2010, 1625–1628

[4] Nephele/PACTs: A Programming Model and Execution Framework for Web-Scale Analytical

Processing

Dominic Battre, Stephan Ewen, Fabian Hueske, Odej Kao, Volker Markl, and Daniel Warneke
In Proceedings of the ACM Symposium on Cloud Computing (SoCC) 2010 ACM, 119–130

[5] Stratosphere Initiative: About Page

<http://www.stratosphere.eu/drupal6/?q=node/2>