Adaptive Resource Management for Distributed Dataflow Systems

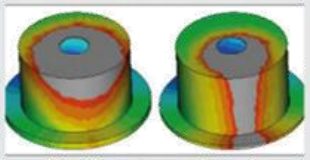
Thomas Renner, Lauritz Thamsen, Odej Kao

Complex and Distributed IT Systems (CIT)

Big Data Applications

Information Marketplaces

E-Health



Materials Science

Scalable Machine Learning

Video Annotation
Text Analytics
Material Characteristics

Statistical Analysis & Machine Learning

Emma (Declarative Specification)

ML | Graph | DataBag

Compiler / Optimizer

SystemML

R - Dialect

Compiler / Optimizer

Apache Flink

Declarative, Scalable Data Analytics

Framework Testing • Development Tools • Benchmarking

Automatic Optimization & Parallelization

XtreemFS | **SDNs** | **Parameter Servers**

Scalable Data Management

Adaptive processing of data- & control flows • Optimization of storage distribution in modern file systems

Motivation

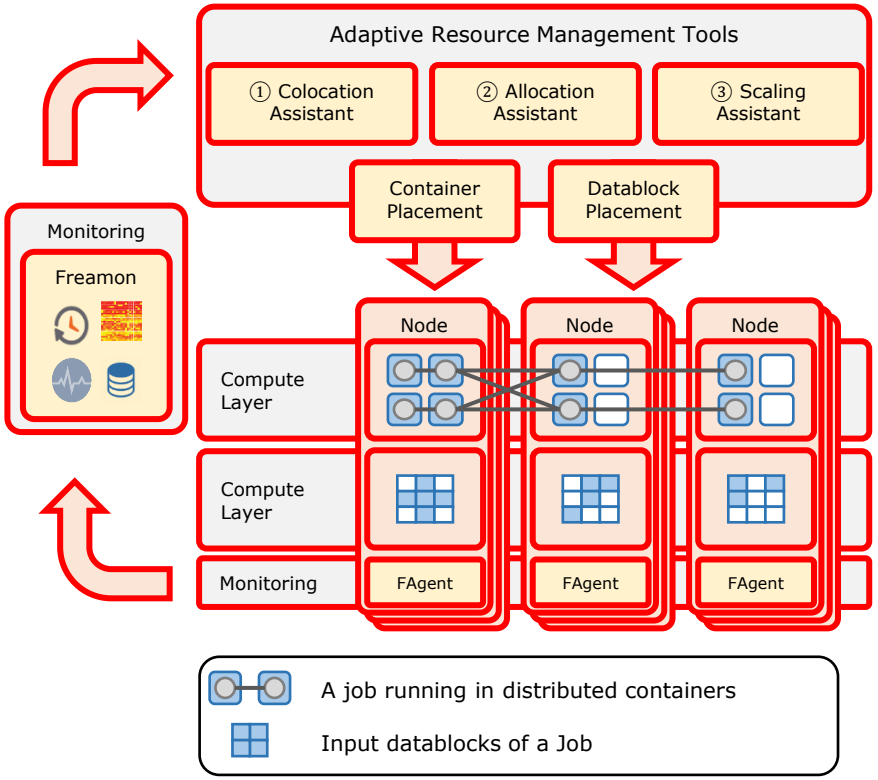
- Distributed analytic jobs stress different resources
  - Machine learning jobs are often CPU intensive
  - Relational data queries are often IO intensive
  - Hard to select resources for specific performance requirements
- Improve integration of Flink for deployments in modern data centers
- Benefit from existing middleware and infrastructure features such as
  - Virtualization (e.g. OpenStack, YARN)
  - Software-Defined Networking (e.g. OpenFlow)
  - Storage Architectures (e.g. SAN, HDFS)

Adaptive Resource Management for Recurring Jobs

- Many jobs are executed repeatedly
  - E.g., daily executed batch jobs and iterative programs
  - Recurring jobs make up to 40% of the jobs in cluster
- Collect job profile information (e.g. runtimes, utilizations)
- Gather user performance demands
- Based on this information, provide scheduling hints for recurring jobs to:
  - Improve resource utilization,
  - Reduce execution time, and
  - Meet performance targets

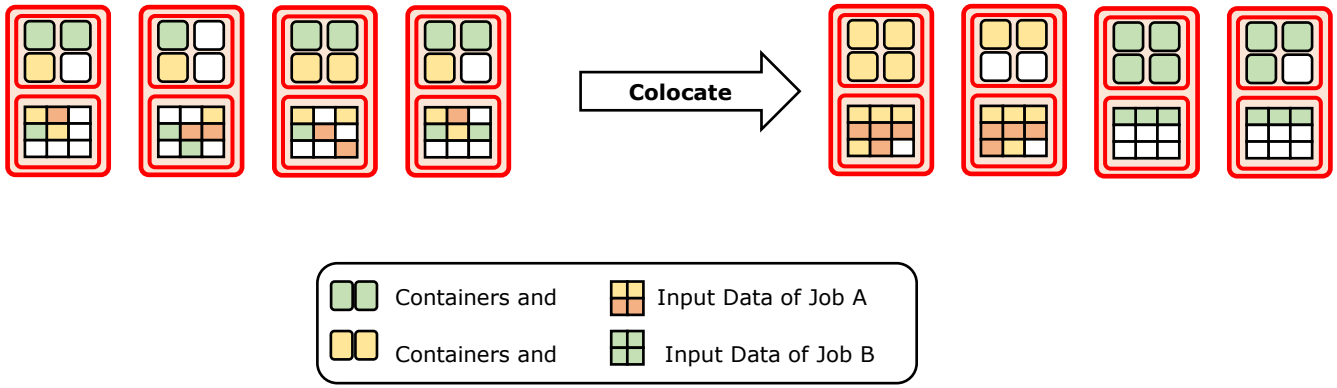
System Architecture

- Job-level cluster monitoring and repository of historical workload data: Resource utilization, job runtimes, data placement, data access etc.



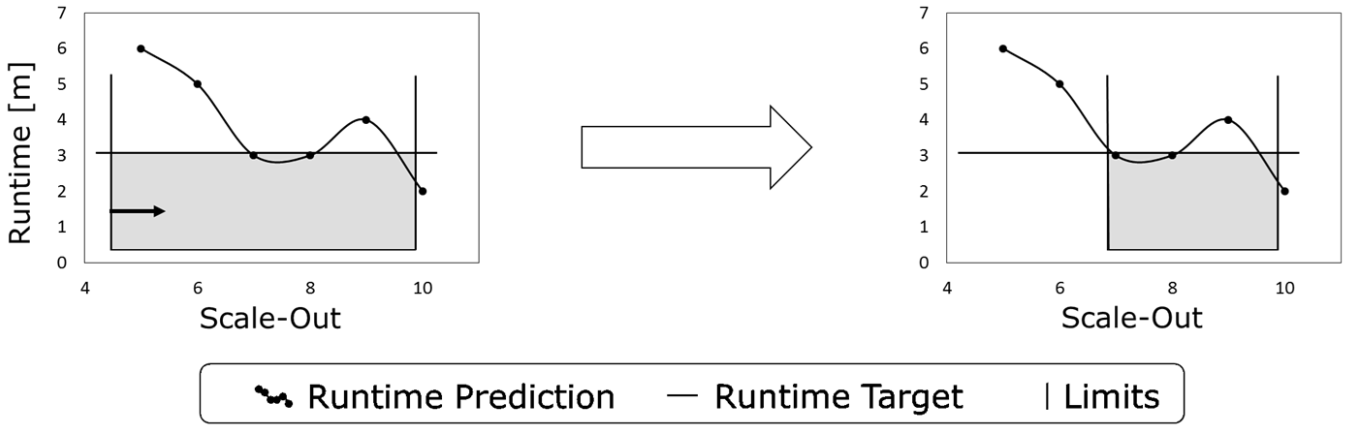
① Colocation Assistant

- Sets of files that are processed jointly are marked as related and automatically colocated on the same set or subset of nodes. Execution containers are also scheduled on this set of nodes. As a result, this improves data locality and reduces the execution time, especially for data-intensive workloads.



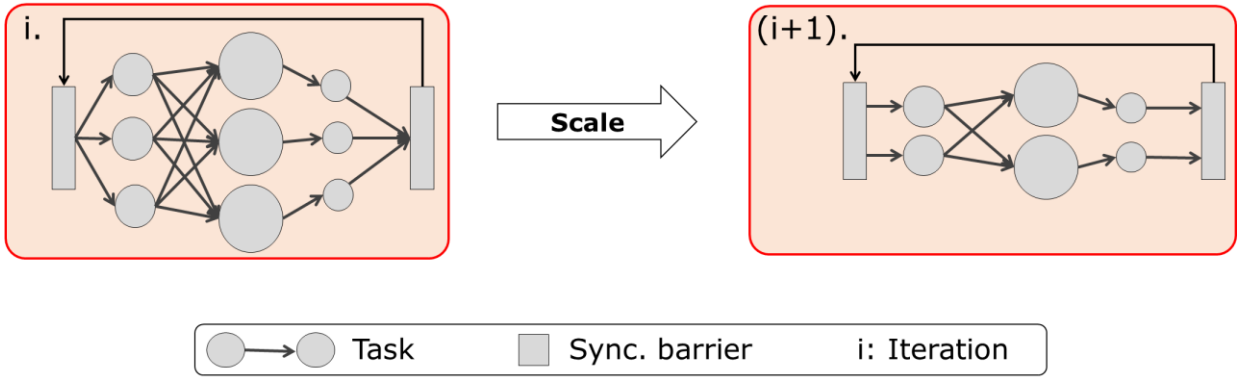
② Allocation Assistant

- Users can explicitly express runtime targets instead of having to guess the required resources (e.g. number of containers, cores, memory, etc.). For this, we model the runtime behavior of jobs using either parametric or nonparametric regression depending on the available historical data.



③ Scaling Assistant

- We dynamically scale iterative dataflow jobs (e.g. many machine learning or graph algorithms) using the synchronization barriers in-between iterations. In particular, we adapt resource allocations towards utilization targets based on collected system statistics.



Adaptive Resource Management for Distributed Dataflow Systems

Thomas Renner, Lauritz Thamsen, Odej Kao

Complex and Distributed IT Systems (CIT)