

Online performance tuning of distributed dataflows using graph analytics

Master Thesis Proposal

Introduction Tuning the performance of distributed dataflow systems like Apache Spark ¹, Apache Flink ², and TensorFlow ³ is a challenging task. Parallel computation is interleaved with data and control communication and execution dependencies typically span multiple system components. To better understand the performance of distributed dataflow computations, the DC-Model group is developing SnailTrail ⁴, an online critical path analysis tool. SnailTrail ingests execution traces from running applications and produces performance summaries that rank activities based on their potential for optimization benefit.

Using real-time execution traces, SnailTrail builds and analyzes a graph model of the execution state called the *Program Activity Graph* (PAG). In the PAG, an edge represents the duration of a worker or communication activity, such as processing, serialization, sending a data message, etc. Currently, SnailTrail slices the execution trace stream into non-overlapping time-based windows and produces a series of PAG snapshots. It then ranks activities by computing a betweenness centrality-based metric on each snapshot.

Thesis Goals The goal of this thesis is to improve SnailTrail's performance summaries by extending its PAG construction and analytics capabilities. In particular, the student will implement and evaluate two additional graph models for PAG construction. First, we will represent the application execution state using sliding windows, thus allowing some overlap between consecutive snapshot analyses. Second, the student will implement a fully-incremental evolving graph model of the application state, such that all execution history is preserved. In order to reveal interesting structural and temporal dependencies among worker tasks in the PAG, the student will then implement a library of centrality and ranking graph algorithms, such as random-walk betweenness centrality [2], random-walk closeness centrality ⁵, load centrality [1], and dominator vertices ⁶. Fi-

nally, the student will evaluate the effectiveness of performance summaries using the proposed metrics and compare them with the existing SnailTrail summaries.

Evaluation and Measurements The evaluation work of this thesis consists of two parts. First, the student will evaluate the implemented graph algorithms in terms of *performance*. As an online profiler, SnailTrail is required to operate with extremely low latency in order to enable real-time performance tuning. At the same time, SnailTrail's graph analytics have to be scalable. SnailTrail should be able to comfortably support the analysis of traces from large-scale deployments with hundreds of parallel workers. Since the PAG is an evolving graph structure, we ideally want to reuse as much computation as possible instead of computing everything from scratch on every snapshot. If time allows, the student will also investigate the incremental implementation and evaluation of SnailTrail's graph analytics library.

The second evaluation part considers the *effectiveness* of the implemented graph metrics. Using Apache Spark, Flink, or TensorFlow, the student will define a use-case and evaluate optimization potential and bottleneck detection capabilities of each newly defined metric on the PAG. The student will investigate whether the metrics reveal causal dependencies among worker activities or other interesting properties of the execution. The goal of this evaluation will be to confirm which of the implemented graph metrics can be used in a practical scenario to tune the performance of distributed dataflow applications.

- [1] Ulrik Brandes. "On Variants of Shortest-Path Betweenness Centrality and their Generic Computation". In: *SOCIAL NETWORKS* 30.2 (2008).
- [2] M.E. J. Newman. "A measure of betweenness centrality based on random walks". In: *Social Networks* 27.1 (2005), pp. 39–54. DOI: 10.1016/j.socnet.2004.11.009.

If you are interested in this project please contact Vasiliki Kalavri (vasiliki.kalavri@inf.ethz.ch). The proposed thesis will be supervised by Prof. Timothy Roscoe.

¹<http://spark.apache.org/>

²<http://flink.apache.org/>

³<https://www.tensorflow.org/>

⁴<http://strymon.systems.ethz.ch/pdf/snailtrail.draft.pdf>

⁵https://en.wikipedia.org/wiki/Random_walk_closeness_centrality

⁶<https://tanujkhattar.wordpress.com/2016/01/11/dominator-tree-of-a-directed-graph/>