Layered Thinking in Vertex Centric Computations

Emanuele Carlini, Patrizio Dazzi

ISTI-CNR
name.surname@isti.cnr.it

Alessandro Lulli, Laura Ricci

University of Pisa
surname@di.unipi.it

Abstract

The Telos framework eases the transition to a vertex-centric approach in the high performance and distributed programming of BigData analytics targeting large graphs. Telos represents a paradigm shift, from “think like a vertex” to “think like a network”.

The recent proliferation of mobile devices and Internet usage has resulted in huge amounts of data. For instance, in 2012, 2.5 exabytes of data were created, every day. This data comes from many heterogeneous sources, including social networks, business transactions and diversified data collections. Industries and academics frequently model this data as graphs in order to derive useful information.

However, it is not always possible to process graphs of such large volumes of data on a single machine. Many different frameworks for large graph processing, mainly exploiting distributed systems, have been proposed in recent years to overcome this limitation.

In order to ease the distribution of the computation across many computers, the vast majority of the proposed solutions exploit a vertex-centric view of the graph [3]. In this approach, the algorithms are implemented from the perspective of a vertex rather than a whole graph. Unfortunately, this shift in the perspective of programmers does not come free-of-charge. Two main issues are identified: performance and adoption.

Performance can be affected by the software design of the programming framework. Moving the viewpoint to a per-vertex perspective needs a careful design of the platform enabling data and computation distribution [1, 2].

The second problem is that programmers may be reluctant to embrace a new paradigm because it will be necessary to adapt classic algorithms to a vertex-centric approach: most
of the existing algorithms must be re-thought or even re-conceived. Solutions targeting this problem aim at providing new tools to help to construct new algorithms.

The Telos framework addresses the adoption issue. Underpinning this framework is the similarity between vertex-centric models and massively distributed systems, for instance P2P. Massively distributed systems commonly rely on a multi-layer overlay network. An overlay can be thought of as an alternative network, built upon the existing physical network, where logical links follow a defined goal. According to Telos, vertices of the graphs can be seen as nodes of the network and edges as links.

We have taken advantage of this similarity to develop three main strategies for large graph processing:

**Local knowledge:** algorithms for overlays are based on local knowledge. Each node maintains a limited amount of information and a limited neighbourhood. During computation, it relies only on its own data and the information received from its neighbourhood.

**Multiple views:** the definition of multi-layer overlays has been a successful trend. These approaches build a stack of overlays, each overlay is characterized by a ranking function that drives the node neighbourhood selection according to a specific goal.

**Approximate solutions:** since overlays are usually based on an approximated knowledge on the graph, algorithms running on them are conceived to deal with approximated data and to find approximated solutions.

Specifically, Telos provides high level API to define multiple overlay views. Telos has been developed on top of Apache Spark. Computation is organized by means of different views of the graph, called Protocol. Some of the most popular massively distributed systems algorithms have been implemented as built-in protocols within Telos. The main task requested to a protocol is to provide a compute function. This function takes as input the messages received by the vertex and the previous vertex state. The contract is to return a new vertex state and messages that must be dispatched to other vertices.

A relevant aspect of Telos is that not only the context of a vertex but also its neighbourhood can change. This functionality is a key part of the Telos framework because it lets users adapt the neighbourhood according to requirements and allows convergence to a graph topology targeted for the problem.

To exploit communication within the neighbourhood of each vertex, three different kinds of communication pattern occur within Telos: (i) intra-vertex to let a vertex access the state of all its layers, (ii) intra-protocol to let a vertex communicate to another vertex on the same layer, (iii) extra-protocol to request the state of another vertex in a protocol different from that operating.

The layered architecture of Telos is shown on the left in Figure 1. A different protocol is executed on each layer. Each vertex has a different state for every layer, as shown in the Telos vertex view on the right.

Telos has been used successfully to improve a state-of-the-art algorithm for the balanced k-way problem and to dynamically adapt the vertices neighbourhood targeting specific problems, for instance, to find similar vertices or for leader election mechanisms.
References

