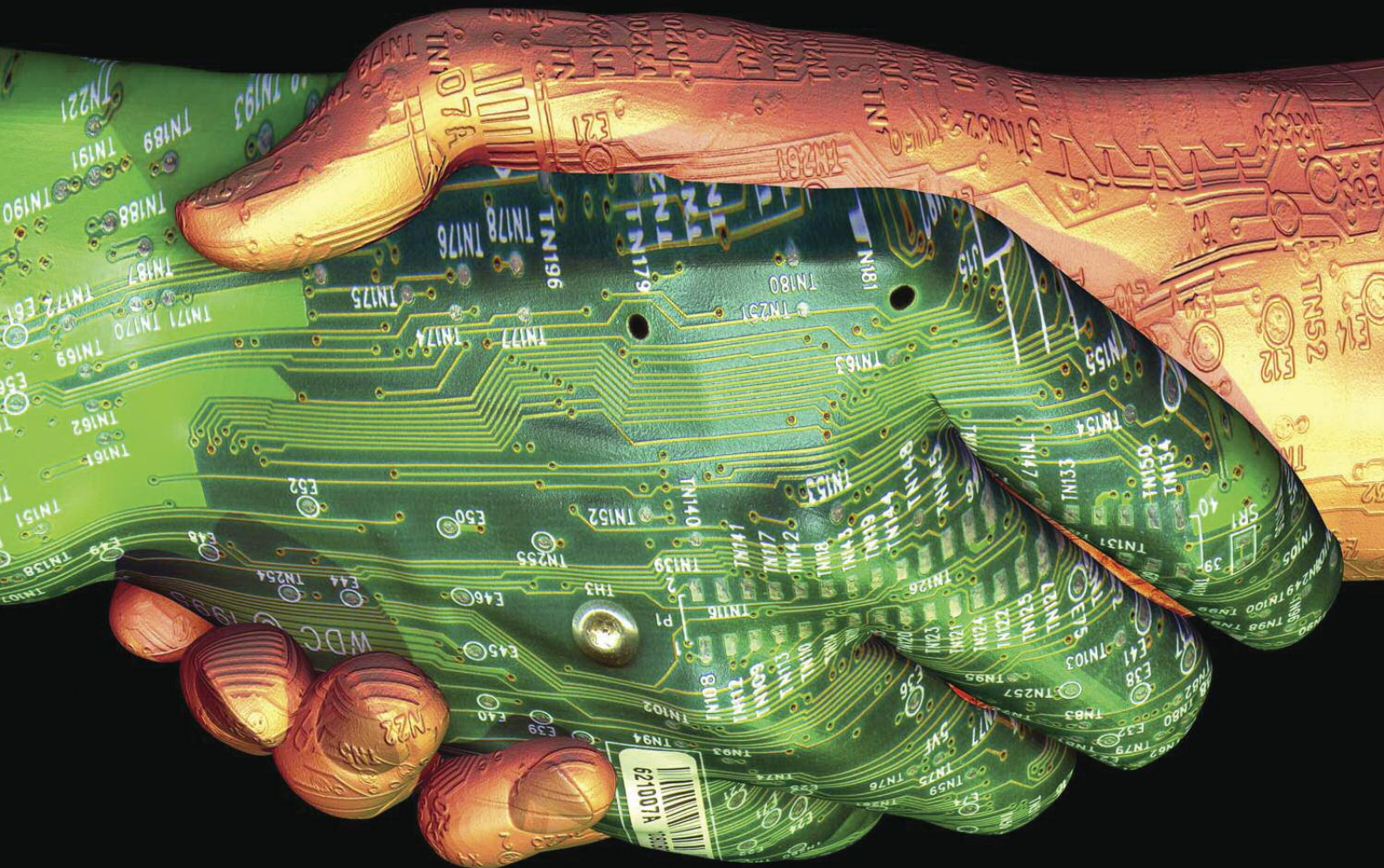


ERCIM NEWS

www.ercim.eu

Special theme: Trustworthy Systems of Systems Safety & Security Co-engineering



Also in this issue:

Keynote:

"Trustworthy Systems of Systems –
A Prerequisite for the Digitalization of Industry",
by Werner Steinhögl, European Commission

Joint ERCIM Actions:

PaaSage and OW2 Announced Platform
Availability on the AppHub Marketplace

Research and Innovation:

Making the Internet
of Things Fly

sequences. The introduction of ICT focusses on reducing the consequences for high frequency incidents, as shown on the right side of the figure. Automation reduces human effort for these incidents because of a reduction in the number of incidents, and possibly also because of automatic restoration processes. However, there is also a change on the other end of the plot. In the absence of preventative measures, automation can lead to larger consequences in low frequency incidents.

The introduction of ICT focusses on reducing the consequences for high frequency incidents, as shown on the right side of the figure. Automation reduces human effort for these incidents because of a reduction in the number of incidents, and possibly also because of automatic restoration processes. However, there is also a change on the other end of the plot. In the absence of preventative measures, automation can lead to larger consequences in low frequency incidents.

This can be illustrated through an example of the restoration process after a power grid failure. More monitoring and controlling devices allow a fast automatic detection and isolation of a failure. The devices also send diagnostics about the precise failure reason and location, which dramatically accelerates the restoration process. Automation reduces the human effort needed to monitor the system. It reduces the required skill set for the repair crews since the system gives more detailed information about its failure. Additionally, it might also reduce the number of repair crews,

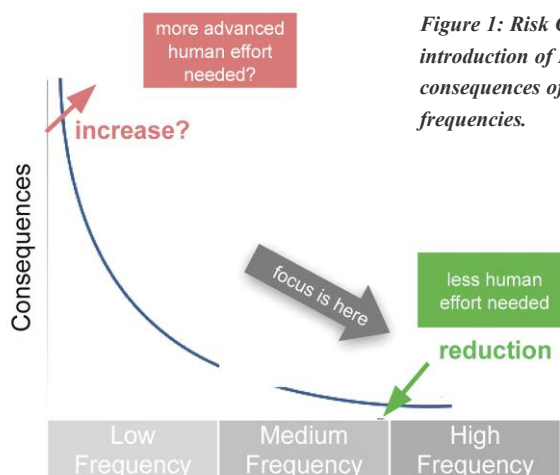


Figure 1: Risk Curve showing how the introduction of ICT may change the consequences of incidents, depending on their frequencies.

as the restoration times are shorter, and owing to better monitoring, a proactive maintenance scheme reduces the number of failures.

However, if the monitoring system fails, the restoration process has to be handled manually again. With a reduced and less skilled repair crew, the consequences of the same outage are bigger. And even more importantly, programming, configuration and operational failures, which are dominant in ICT systems, add additional failures and may lead to very unpredictable states of the system and are more difficult to locate and restore.

In summary, the introduction of automation may have unwanted effects for low frequency incidents. This can be circumvented by the following endeavors: first, by using the saved human effort in normal operation to cover less frequent incidents; second, by increasing the skill set for operational staff to cover new failures and rare

events; third, by keeping the staff trained to a high standard and having efficient and well-established processes to deal with rare events.

References:

- [1] International Energy Agency (IEA), "Technology roadmap: Smart grids," http://www.iea.org/publications/freepublications/publication/smartgrids_roadmap.pdf, 2011.
- [2] P.Cholda et al., "Towards risk-aware communications networking," *Rel. Eng. & Sys. Safety*, vol. 109, pp. 160–174, January 2013.
- [3] S. Rinaldi et al., "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Systems*, vol. 21, no. 6, pp. 11–25, Dec. 2001.

Please contact:

Jonas Wäfler, Poul E. Heegaard
 NTNU, Norway
 E-mail: jonas.waefler@item.ntnu.no,
poul.heegaard@item.ntnu.no

Layered Thinking in Vertex Centric Computations

by Emanuele Carlini, Patrizio Dazzi, Alessandro Lulli and Laura Ricci

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 pro-

cessing, 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 [1]. 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 [2][3].

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

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.

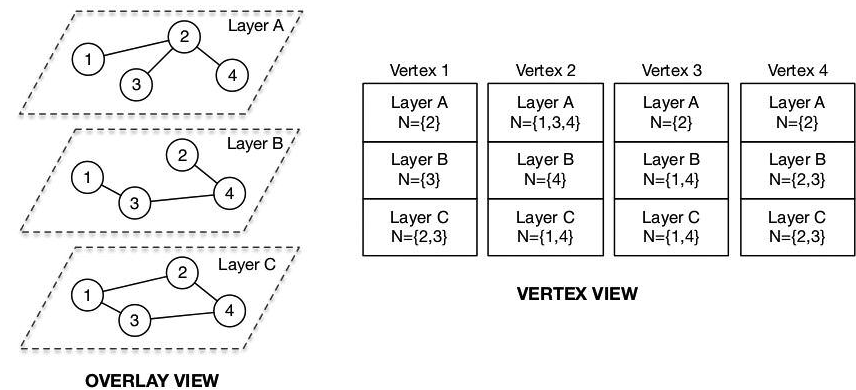


Figure 1: Layered architecture and interactions.

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 dif-

Links:

Telos API:

<https://github.com/hpclub/telos>

References:

- [1] R. R. McCune, T. Weninger, G. Madey: "Thinking Like a Vertex: a Survey of Vertex-Centric Frameworks for Large-Scale Distributed Graph Processing."
- [2] E. Carlini, et al.: "Balanced Graph Partitioning with Apache Spark, in Euro-Par 2014: Parallel Processing Workshops (pp. 129-140). Springer, 2014.
- [3] A. Lulli, et al.: "Cracker: Crumbling Large Graphs Into Connected Components", 20th IEEE Symposium on Computers and Communication, ISCC2015.

Please contact:

Emanuele Carlini, Patrizio Dazzi
ISTI-CNR, Italy
E-mail: emanuele.carlini@isti.cnr.it,
patrizio.dazzi@isti.cnr.it

Alessandro Lulli, Laura Ricci
University of Pisa, Italy
E-mail: lulli@di.unipi.it,
ricci@di.unipi.it