Towards Computing Without Borders: A Data Processing Plane
Reginald Cushing¹, Marian Bubak¹,², Adam Belloum¹, Marc X. Makkes¹, Cees de Laat¹
¹Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH, Amsterdam, Netherlands
²AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow

Motivation
Data partitionability, processing complexity and locality play a crucial role in the effectiveness of distributed systems. Through virtualization, resources have become scattered, heterogeneous, and dynamic in performance and networking. Collective and collaborative use of these resources for data processing is our main challenge.

Automata as a Data Processing Schema
Automata is an intuitive way to describe data processing as a transformation from one state to the next. The data transformation model can be considered as a 5 tuple NFA:

\( (Q, \Sigma, \delta, q_0, F) \)

- \( Q \) is the set of states the data object can be in.
- \( \Sigma \) is the set of functions that performs the data transformations.
- \( \delta \) is the transition function that maps data and functions to new states such that: \( Q \times \Sigma \rightarrow P(Q) \).
- \( F \) is the set of final data states which mark the completion of processing.
- \( q_0 \) is a starting data state.

Background
In eScience, coordinating multiple tasks for running in-silico experiments is often the realms of Scientific Workflow Management Systems (SWfMS). These are often centralized systems that work in confined resources.

A common denominator in most workflow systems is that the unit of reason is the process i.e. the abstract workflow describes a topology of tasks configured in a certain way. This is often tailored to the underlying infrastructure. Thus the process ordering is a description of how to best exploit resources and not necessarily a description of data processing.

The complexity and dynamism in big data processing entails a new unit of reason: the data itself. An abstract model for data processing will solely describe data transformations agnostically from the underlying resources.

Distributed Data Processing as a Protocol
Automata data model describes the abstract data processing model. The same model is used to build a distributed processing infrastructure around the data processing schema. The schema represents the knowledge of how data can be processed which at a network and resource level this represents a data routing table.

Globally distributed resources are combined together in the PUMPKIN framework through a data processing protocol. Data is partitioned into packets. A packet is an atomic unit of data processing. Each data packet can encapsulate the automata as part of the header. The automation header makes the packet self aware of where it has to go for processing. The data packet can also contain the code for processing the packet.

Data Packet = Data + Automaton + Code + State

Each node in PUMPKIN discovers routes to other nodes. A routing table allows nodes to send data packets to the next node in a P2P fashion. In SDNs, the routing table can be used to reconfigure the network.

PUMPKIN in Action

1. Function network (workflow) corresponding to the data automata in 2 for both applications.

2. Data automata for 2 applications co-hosted on the network: A Twitter sharing application and a bio-medical application.

3. Network connections for connecting VMs from different providers. As can be noted some VMs are hosting functions from both applications.

4. Live processing and networking statistics from individual processing functions.

References

Acknowledgments: This work is funded by the COMMIT (http://commit.nl/) project.