13 Feb 2020

Location, Location, Location:
The Exposed Buffer Approach to Problems of Data Logistics

Micah Beck, University of Tennessee
Martin Swany, Indiana University

One of the things that gets in the way of accurate reasoning about computer systems is dealing with location. As long as we don't consider the location of a resource among its relevant characteristics, we can draw highly abstract conclusions about its behavior. A classic example is the use of variables in Boolean algebra to represent wires in combinational circuits. The number and location of sensors relative to the drivers of a signal determines many practical characteristics of the circuit, but strictly logical reasoning allows these to be ignored.

It is a common goal that users of distributed resources should be allowed to express their intentions without specifying the location of resources and without having to take into account the implications of resource allocation. Such systems are left with a difficult task, to map such abstract intentions to the concrete resources and activities supported by actual systems. The quest to bridge this gap has challenged Computer Scientists for decades.

All too often, the designers of computer systems have attempted to finesse this problem by creating systems in which it is claimed that location is irrelevant. There are two main approaches:

1) designing resource placement and scheduling algorithms that minimize the effects of locality. and
2) provisioning communication, storage and processing ubiquitously to minimize the effects of location.

The goal of algorithmic optimization is ambitious, and is often offered at the start of the design cycle. However if optimization proves insufficient, overprovisioning combined with a decision to ignore the residual effects of locality are common fallbacks. In the past Moore's Law and Dennard Scaling have provided much-needed relief to the field, but over time the cost and complexity of ignoring locality always assert themselves. Such is the case with the management and processing of huge data sets today.

The attractive force of location-independence is so great that it can tear communities apart. Sometimes the specific needs of a subcommunity allow for the use of specialized solutions to solve the general problem of mapping location-independent intentions to location-specific plans. In such cases the subcommunity may direct economic power to its specialized solutions so forcefully that the needs of others are neglected. Arguably this is the case in the molding of the World Wide Web, the Domain Name System and Internet topology to the needs of commercial Web service providers including Content Delivery Networks and Cloud infrastructure providers. The needs of the scientific community are not even secondary; they are all but ignored.

The unwillingness or inability to hide locality through optimization commonly results in the default position of exposing location directly to end users, creating a burden of complexity which depends on information that they may not possess (e.g., "Choose a mirror site near you"). Perhaps failure to take account of locality algorithmically leads to avoidance. Another common response is to rely on unrealistic levels of provisioning, with the result that the community of possible deployment is restricted to an elite.

An alternative approach is to address the lack of effective universal optimization by defining a uniform design space and then supporting heterogeneous solutions for particular application communmities. This means exposing the tools for placement of primitive resources and monitoring of dynamic aspects of topology (such as resource utilization and reliability) to a programming layer that can create solutions that meet the needs of important but not hegemonic application communities.

13 Feb 2020

Logistical Networking (LN), under development as an architecture and a tool stack (http://data-logistics.org) for over 20 years at the University of Tennessee, Indiana University and Vanderbilt University takes the approach of exposing a primitive, limited and best-effort buffer management service with a uniform interface as a virtualization of the local storage and processing resources of the local node, i.e., a converged node OS. While network links are implemented using TCP/IP, that is an implementation convenience rather than an architectural preference. The intent is that necessary resource location and use information (generalized topology) is visible to every intermediate node and endpoint, and that connection topology may be made irregular by the interposition of NATed routers, firewalls, security policy or the use of LAN or other non-IP links.

The LN stack, implemented as in the Data Logistics Toolkit, is supported by Martin Swany's research group at Indiana University's Intelligent Systems Engineering Department. It includes an implementation of a fundamental LN storage and buffer management server called the Internet Backplane Protocol (IBP) "depot" developed by Alan Tackett of Vanderbilt's Advanced Computing Center for Research and Education (ACCRE). Experimental work at Tennessee and Indiana has validated approaches to computation and programmable networking based on the deployment of limited computational operations extending IBP. This "in-locus" model will be supported in the future by the Data Logistics Toolkit.

The DLT has been deployed on a number of shared NSF-funded experimental testbeds, including PlanetLab and GENI. The Research and Education Data Depot Network supported 6 PB of storage distributed throughout the US R&E network and at CERN, moving and storing Particle Physics data from the Large Hadron Collider and from the Earth Observing Data Network, an LN-based open content distribution network that carried satellite images including MODIS and other USGS-operated satellites.

The same depot and the interoperable IBP protocol are used as the basis of a second storage stack, developed and supported by Vanderbilt's ACCRE, which implements the Lstore enterprise storage service used to manage multiple petabytes of data, including significant collections of video and physics data (https://www.vanderbilt.edu/accre/sc19/lstore/), At the University of Tennessee, experimental versions of in-locus computation have been used to implement a wide variety of functions including in-network compression, merging of sorted streams, distributed visualization and search. A recent version developed at IU, called InLocus, targets extremely efficient stream processing at the edge.

A current effort in DLT-based infrastructure is a proposed computational facility with a large central cluster built out of IBP depots equipped with in-locus computational power which is allocated in a manner based on fair contention within an authorized user community (a computational analogy to TCP friendly contention for bandwidth). Parallel data storage and access will be provided by Vanderbilt's Lstore. The facility will also include smaller regional computational clusters built using IBP depots and depots will also be distributed throughout the core of the R&E network and at edge locations where instruments, sensors or simulations are generating massive data flows. All elements in this distributed platform will be capable of at applying least limited computational power to implement data reduction, selection and intelligent routing as well as the application of neural network models that have been trained using the large scale and perhaps more centralized clusters. All of the components of this facility will be built out of interoperable exposed buffer services. The distributed services, data movement and replication will be orchestrated by Indiana University's Intelligent Data Movement Service. The targeted application domain is on-demand, best-effort analysis of massive data sets and visualization for the national scientific research community.