Glimpsing a Yottascale Data Ecosystem when the Fog Lifts

Micah Beck (mbeck@utk.edu), Terry Moore (tmoore@icl.utk.edu)

As the BDEC Pathways report[1] and other contemporary sources make clear, in the new era of data intensive science and engineering, nearly every field confronts formidable problems of "data logistics," i.e., of the management of the time sensitive positioning and processing of data relative to both its intended users and the public or private resources available to them. Since there is widespread consensus that the volume of digital data worldwide is increasing exponentially, with plausible projections putting the total at around 20ZB by 2020, this situation is bound to get worse. Given the correlated, explosive growth in prolific data generators, ranging from the Internet of Things (IoT) and mobile devices, to scientific, engineering, medical, military and industrial equipment and infrastructure of all kinds, the reality of a Yottascale (10^{24}) digital universe is just over the horizon. Creating a widely shareable infrastructure that can provide the computing, storage/buffer, and communication services required to support data logistics for data ecosystems at that scale involves challenges that are obviously formidable, to say the least.

Beyond the notorious five V's of data—volume, velocity, variety, variability, and value—what makes this problem especially "wicked" is the fact that, in a world of utterly pervasive IoT, mobile devices, and cyber-physical systems, *the data producers are everywhere* and this means, in turn, that the nodes of our general purpose Advanced Cyberinfrastructure Platform (ACP), our fabric for the "data periphery," will have to be everywhere too. We know that ACP nodes (in a wide range of sizes and capabilities) will have to be deployed ubiquitously in the data periphery because at least some important applications will require low latency response, or local data reduction, or high confidence security/privacy, or highly survivable services, etc. So when we think about a future defining ACP, we are necessarily thinking about something with the same kind of boundary defying footprint as the Internet, with nodes deployed at and interoperable across every level of the ecosystem: wrist, pocket, purse, car, home, farm, office, building, campus, town, city, region, nation, globe. The question is "How can we create an ACP with that kind of deployment scalability?"

As we all know, the conventional answer to this question today is to use Virtual Machines and Containers to move miniaturized versions of the Cloud data center model into nodes located throughout the network: the Cloud is supposed to roll out across the data periphery as Fog computing. This model is characterized by persistent processes and file or database systems that allocate resources indefinitely on specific nodes. It further uses embedded state to implement high level services, even though creating sufficiently flexible service/state management functions for them (e.g., migration and fault tolerance) is difficult. But the model has been extremely successful inside commercial cloud data centers and content delivery networks, where problems with automated management have been countered by concentrating on application domains that generate a lot of income. It has been so successful in fact that the prevailing opinion in the ICT community is that no other model is viable for the great data periphery: technological path dependence and overwhelming commercial power dictate an inescapable destiny, regardless of the experience and principles of "Computer Science."

The authors of this position paper have significant doubts about this dominant consensus. In view of the history of distributed systems over the past three decades, the fundamental problem we see is that any ACP that converges over the top of the conventional "three-silo" model—process+file system+Internet—will prove too complicated to permit scientific applications at scale (especially those that incorporate the data periphery) to be highly automated, that is, automated in the way the Internet has been. This is illustrated, for instance, by the fact that CDN's are relatively expensive to run, so that the USGS cannot afford to pay the cost of making all its satellite data publicly available without throttling bandwidth. If we believe plans for the future of scientific cyberinfrastructure are fated to collide with the silos in the Fog or at the Edge, then perhaps some, even many, of the applications research communities will want to deploy have requirements that are simply ruled out.

We believe that designing an affordable and sustainable ACP for the extreme scale data ecosystems of science's future, an ACP that rules in as many applications as possible, is *a grand challenge problem* that calls for a radically different model from the current conceptions of Fog and Edge computing. We can understand why the problem of designing such an APC is so formidable by considering four plausible design constraints. A scalable and sustainable ACP must

- *Combine global interoperability with diverse, community specific policies*: Service definition interoperability is widely agreed to be fundamental to the success of a sustainable ACP. But as BDEC Pathways report reminds us, "As a practical matter, real interoperation is achieved by the definition and use of effective spanning layers,"[2] i.e. a common service interface that aggregates access to heterogeneous resources in support of a generalized set of applications that need to use such resources. If the ACP's global services layer (e.g., layer three in the Internet model, see Figure 1A) is also the global spanning layer (i.e.," the narrow waist of the hourglass"), then any use of the system must go through that interface, and therefore must be open to all system users. However, the current Internet shows that this approach leaves the system wide open to all manner of malicious ingenuity by bad actors, making acceptable—let alone community specific—security, privacy, and federation policies extremely difficult to achieve. If we want a shared, pervasive ACP that provides global interoperability, it is doubtful that building on the current Internet silo (see Figure 1B) will get the community to that goal.
- *Manage tradeoffs between node autonomy and manageability*: Logical centralization characteristic of current Fog models enables focused "command and control" of resources, but can suffer from bottlenecks, latencies and other data logistics issues due to the separation between the policy and processing planes. On the other hand, physical distribution enables local control, but makes global optimization (e.g., for performance and reliability) and resource allocation difficult. The design of current operating systems and networks conflate the placement of resources with their control. Can we design distributed systems in a way that provides appropriate performance and reliability in widely different environments by adapting the separation of data and control in diverse and flexible ways?
- *Create a topologically enabled interface to system resources for good data logistics*: Effective and efficient management proximity/locality/logistics in a shared ACP demands 1) awareness of where the data and the resources to control and process it are, and 2) some control over when, where and in what form it goes next. The Internet's spanning layer hides topology from higher layers for good reasons, but in doing so it restricts their ability to perform data logistics optimizations. Can we achieve necessary tradeoffs between abstracting away from topology and heterogeneity serving multiple client communities by moving the spanning layer below layer 3?
- *Future-proof the design to maximize its sustainability*: If a shared ACP is going to be sustainable, it will have to be able to absorb successive waves of innovation in the hardware substrate it runs on and in the applications that people want to run on it. Two factors combine to make this an extremely difficult problem. On one hand, network effects inevitably make the spanning layer of any universal bearer platform a point of design ossification, inhibiting future innovation at that layer [3]. On the other hand, any standard tends to restrict heterogeneity at the layer at which it is imposed. Taken together, these two facts imply that the lower the spanning layer is placed, the more sustainable the infrastructure will be, since a lower layer standard can allow many choices at the layers above it, whereas a high level standard locks in choices that it inherits from the limitations of the silo layers that it relies on. This poses a fundamental problem for designing a sustainable ACP with the common service interface at a very low layer: how can the resources of its intermediate node be modeled in a manner that is general enough to support all necessary services but still be simple, generic and limited portable to almost any hardware substrate?

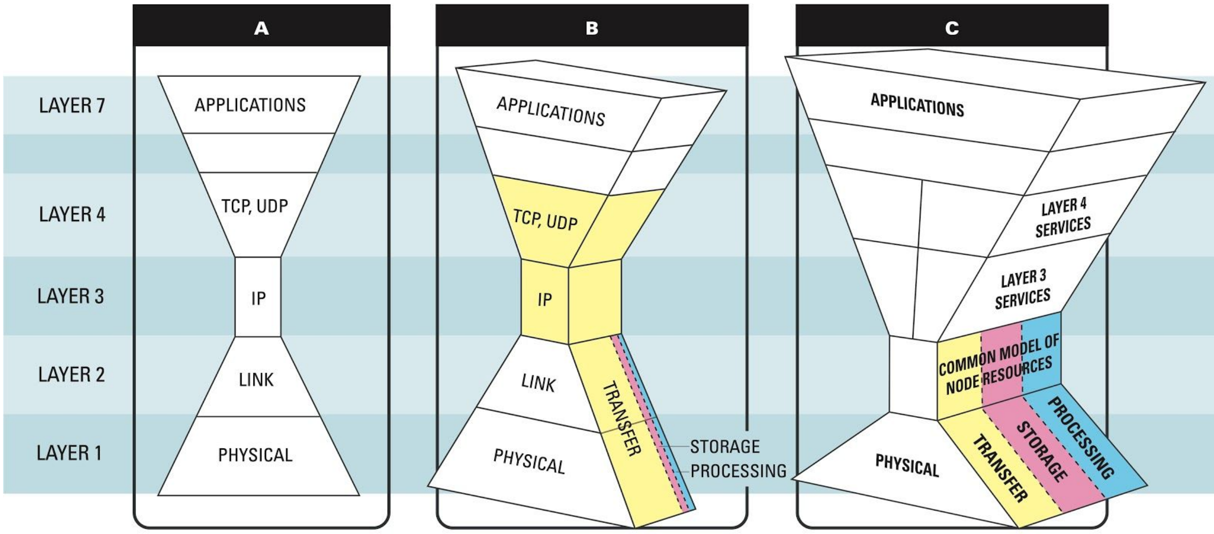

Figure 1: Moving the common interface from layer 3 of the Internet silo (B) to a lower layer (C) that exposes all its basic resources.

What we are proposing is to make the model of the intermediate node—the node operating system, if you will—the basis of interoperability (see Interoperable Convergence of Storage, Networking and Computation, FICC 2019, [4]). This means exposing a standards-compliant platform on which layer 3 services can be built, establishing interoperability below that layer, but not by adopting a uniform model of local networking. Instead the focus moves to the intermediate nodes themselves, virtualizing all of their local services including communication between "virtually adjacent" nodes. The fundamental abstraction on which these local services are built is the memory buffer or storage block, which is the common building block of operating system services.

If interoperability is moved to the resources of the intermediate node then the services built at "layer 3" need not be just those that are usually considered to be "communication" but can be much more general (see Figure 1C). For instance, a CDN can be a layer 3 service, but so can a distributed file system or a distributed facility that implements computing and caching of results in some particular domain of interest. The idea is to create a platform that can support distributed services in a fundamentally different way than the Internet approach (i.e., using stateless transmission to tie together servers located outside the network core) and not just a variant that loosens up the emphasis on "stateless" and "transmission".

The goal is to deploy nodes that support data transfer, persistence and transformation while being operated in a manner similar to the network, by engineers that monitor and adjust configuration but not by system administrators that get involved with the applications and their users imposing overhead and requiring fallback to complex manual procedures. This is the area where Data Logistics research has something unique to offer. Then there is a lot of engineering and further research to do in realizing the architectural vision.

[1] M. Asch *et al.*, “Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry,” *Int. J. High Perform. Comput. Appl.*, vol. 32, no. 4, pp. 435–479, 2018.

[2] D. D. Clark, “Interoperation, Open Interfaces, and Protocol Architecture,” in *The Unpredictable Certainty: Information Infrastructure through 2000*, Washington, DC, USA: National Academy Press, 1997, pp. 133–144.

[3] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet impasse through virtualization,” *Computer*, vol. 38, no. 4, pp. 34–41, 2005.

[4] Micah Beck, Terry Moore, Piotr Luszczek, and Anthony Danalis, “Interoperable Convergence of Storage, Networking, and Computation,” presented at Future of Information and Communication Conference (FICC) 2019, to appear. (https://arxiv.org/abs/1706.07519)