EdgeNet: A Global Cloud That Spreads by Local Action

Justin Cappos
NYU Tandon School of Engineering
NYU
New York, NY
jcappos@nyu.edu

Matthew Hemmings
US Ignite
Victoria, Canada
discount.yoyos@gmail.com

Rick McGeer
US Ignite
Orinda, CA
rick.mcgeer@us-ignite.org

Albert Rafetseder
NYU Tandon School of Engineering
NYU
Vienna, Austria
albert.rafetseder@univie.ac.at

Glenn Ricart
US Ignite
Salt Lake City, UT
glenn.ricart@us-ignite.org

I. Introduction

EdgeNet is a distributed edge cloud, in the family of PlanetLab[1], GENI[2], Canada's SAVI infrastructure[3], Japan's JGN-X[6], Germany's G-Lab[5], and PlanetLab Europe. EdgeNet is designed with the experiences of its predecessors in mind, and has chosen design features to avoid the pitfalls that befell earlier systems. Moreover, edgeNet takes advantages of recent technological advances unavailable to its predecessor systems.

EdgeNet is a software-only infrastructure. Previous infrastructures all used dedicated hardware, which had three major negative effects.

- Scalability was limited. PlanetLab eventually achieved 700 sites, but maintenance of those sites was a full-time occupation for a team of six, largely troubleshooting node failures.
- Hardware refresh formed a major expense for these systems, since about ⅓ of capital costs are required annually for hardware refresh.
- Local system administrators found maintenance of special-purpose hardware a burden.

A. Why an Edge Cloud?

EdgeNet is the platform for a new class of applications and services: Cloud-in-the-loop systems. Lightweight devices (smartphones, sensors, actuators) interact with the real world (things and people) and can do some lightweight computation, but their computational power is soon exhausted. Conversely, Cloud systems are arbitrarily powerful, and so the natural pairing is lightweight devices with Cloud systems: devices offload storage and computing to Cloud nodes.

However, Cloud systems are sparse; Google Cloud has 11 Points of Presence in North America in five regions (a sixth is planned). Distance is time; device-Cloud latency is on the order of tens of milliseconds or more. This means that a Cloud node and a device can have at most a few transactions per second, severely limiting the classes of applications which feature device-Cloud interaction. Reducing the latency between device and Cloud opens up a broad array of new services, including data-intensive visualizations, virtual reality on thin devices, prediction and analysis for the Internet of Things, including real-time video processing and analytics.

B. Why Software-Only?

EdgeNet nodes are VMs which run on general-purpose computing nodes; these may be local Clouds, servers, or even personal or embedded computers running the appropriate software. EdgeNet is software-only because edge Clouds must be orders of magnitude larger than existing commercial Clouds. This is due to a theorem from physics and plane geometry. Getting a Cloud node within k milliseconds of a point on a plane requires that the Cloud node must be within k/c meters of the point, where c is the speed of light. Thus, getting a Cloud node within k milliseconds of anywhere requires covering the plane with circles of radius (k/c). The area of a circle is proportional to the square of its radius, so this amounts to covering the plane with circles of area $(k/c)^2$. Since the area of the plane is unchanged, reducing k by a factor of x requires increasing the number of circles (and thus the number of POPs) by a factor of x^2. Cutting latency by a factor of 10 means 100 POPs for every Cloud POP today. This is an enormous and slow undertaking, if done as a Cloud with dedicated hardware and bespoke software.

However, there is another model for building infrastructure. In the late eighties and early nineties, digital libraries appeared to he a massive undertaking, with enormous disk farms and city-scale computing. There was no centralized, massive buildout; rather, a simple piece of software was made available for download by the National Center for Supercomputing Applications. Within a decade, the World Wide Web had spread to millions of sites and created a digital library of massive, worldwide scale by the local, individual action of millions of individuals and organizations.

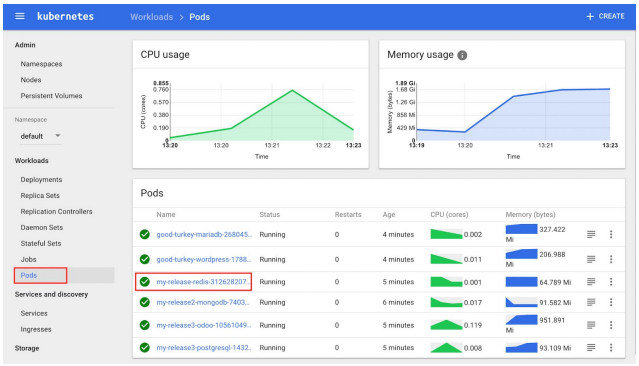
EdgeNet's vision is to replicate the experience of the World Wide Web: to build a world-girdling, always-available, next-to-you-wherever-you-are Cloud through the actions of millions of individuals and organizations.

II. EdgeNet Technology

EdgeNet is based entirely on industry-standard software: Docker[7] as the containerization solution and Kubernetes[4] as the node and cluster manager. Previous infrastructures used special-purpose control frameworks, leading to both a training and a maintenance burden.

EdgeNet is a global Kubernetes cluster. In this, it is unique: Kubernetes is designed to be run on a cluster with low inter-node latencies; in contrast, edgeNet's inter-node latencies are on the order of tens of milliseconds. A edgeNet worker node is a standard Ubuntu virtual machines in a local Cloud, or on a physical machine, on a part or full-time basis. A site downloads the edgeNet Worker node image, a standard Kubernetes node, and then runs a simple pre-installed script to attach to the global Kubernetes cluster.

EdgeNet is designed to be as thin an overlay on Kubernetes as possible. Once a user is accredited to the EdgeNet system and downloads his configuration file, he is presented with a Kubernetes dashboard like this one:



The experimenter uses the dashboard and/or the Kubernetes tool kubectl to instantiate and manage container swarms ("pods") across the distributed cluster, exactly as he currently does on a centralized Kubernetes cluster.

The intended audience and users of EdgeNet are distributed systems developers; these developers find its capabilities a substantial improvement over previous infrastructures such as PlanetLab and GENI. In particular, they will find it spreads far wider and faster than previous fixed infrastructures; has far the ability to instantiate and compose service instances, where each service instance is an ensemble of microservices; and has much greater ability to offer mobility and robustness under node and network failure.

III. Mapping Distributed Systems Concepts to Kubernetes

Usage of an existing technology such as Kubernetes requires mapping current distributed systems concepts to native concepts. A distributed systems *slice* maps naturally onto the Kubernetes concept of a *ReplicaSet*/*DaemonSet*; a sliver maps naturally onto a *Pod,*

A. Experimenter Isolation

Experimenters are isolated from each other via Kubernetes' role-based access control. When an experimenter gets an EdgeNet account, he is assigned his own *namespace*. All interactions with EdgeNet are under this namespace. Namespaces are isolated; users in one namespace cannot see the

B. Node Selection

A concept not native to Kubernetes is user selection of sliver placement; since Kubernetes is primarily designed to run on a colocated cluster, the scheduler is free to assign pods to nodes. However, Kubernetes does provide a hook to permit users to place their own slivers: *labels*. When an experimenter allocates a slice, he can choose a label for that slice arbitrarily, and assign nodes to that label. He can then specify that label as the place where pods can be assigned; by choosing the number of replicas as the number of nodes with the assigned label, the scheduler is forced to assign labels as the experimenter directs.

IV. Status

EdgeNet is up and running at 35 sites across the US, Canada, and the EU. It can be used at https://www.edge-net.org. Not all of the planned features (notably labels as node selection) are as yet implemented, but they will be well before the conference. Nodes can be added to the system within 15 minutes; a "Hello, World" introductory service deployed across the system in 5.

References

[1] Larry Peterson, Andy Bavier, Marc E. Fiuczynski, and Steve Muir. 2006. Experiences building PlanetLab. In Proceedings of the 7th symposium on Operating systems design and implementation (OSDI '06). USENIX Association, Berkeley, CA, USA, 351-366.

[2] The GENI Book (2016), doi:10.1007/978-3-319-33769-2 edited by Rick McGeer, Mark Berman, Chip Elliott, Robert Ricci

[3] Leon-Garcia, A & Bannazadeh, Hadi. (2016). SAVI Testbed for Applications on Software-Defined Infrastructure. The GENI Book. 545-562. 10.1007/978-3-319-33769-2_22.

[4] What is Kubernetes? https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

[5] Müeller P., Fischer S. (2016) Europe's Mission in Next-Generation Networking with Special Emphasis on the German-Lab Project. In: McGeer R., Berman M., Elliott C., Ricci R. (eds) The GENI Book. Springer, Cham

[6] Nakao A., Yamada K. (2016) Research and Development on Network Virtualization Technologies in Japan: VNode and FLARE Projects. In: McGeer R., Berman M., Elliott C., Ricci R. (eds) The GENI Book. Springer, Cham

[7] Docker. https://www.docker.com