Cresco

THE PROBLEM

- Technical constraints (speed of light, computational, storage, throughput, power, etc.) prohibit the movement and processing of data remotely.
- There exist a data-value paradox, where information that is required to model data value has an indeterminate collection rate and duration. This is to say that the value of data is not known until the cost of transmission and processing have been incurred.
- Privacy, legal, and operational constrains preventing remote data exchanges.

CRESCO EDGE COMPUTING FRAMEWORK

We present a new distributed agent-based framework designed to address the challenges of edge computing. This actor-model framework implementation is designed to manage large numbers of geographically distributed services, comprised from heterogeneous resources and communication protocols, in support of low-latency real-time streaming applications. As part of this framework, an application description language was developed and implemented. Using the application description language, a number of high-order management modules have been implemented including solutions for resource and workload comparison, performance observation, scheduling, and provisioning.

ADVANTAGES OF CRESCO

Cresco provides a number of advantages to those trying to adapt their systems to the edge computing paradigm, such as:

- Inter-agent control and data plane communications channels:
    - **Secure**: Channels have SSL encryption enabled at each level of the Cresco hierarchy, using a unique encryption signature per connection
    - **Fault-Tolerant**: During any loss of connectivity, the agent will attempt to re-establish communications with a nearby agent with which is it authorized to communicate, or a specific IP if a rigid Cresco hierarchy must be maintained
    - **Low-Latency**: Cresco utilizes robust, efficient proven communications libraries to ensure reliable, low-latency communications are always available
    - **Multi-Protocol**: An added advantage of the communications libraries is the ability to support a number of communication protocols such as AutoWire, MQTT, OpenWire, REST, RSS and Atom, STOMP, WSIF, Websockets, and XMPP
- An application deployment language is available (Cresco Application Description Language, or CADL) to allow for the quick deployment of highly complex, multi-agent applications
    - Server-less management of resources
    - Global scheduling and optimization of resources
    - Prediction of future resource needs
    - Dynamic recovery/rescheduling of services
- Encapsulated code execution that allows on-the-fly deployment of required routines without any downtime
- Runs on anything able to run a standard Java JVM