ALTO Use Case: Resource Orchestration for Multi-Domain Data Analytics

draft-xiang-alto-exascale-network-optimization-03

Justas Balcas\textsuperscript{2}, Greg Bernstein\textsuperscript{3}, Haizhou Du\textsuperscript{4}, Azher Mughal\textsuperscript{5}, Harvey Newman\textsuperscript{1}, Qiao Xiang\textsuperscript{6}, Y. Richard Yang\textsuperscript{6}, Jingxuan Zhang\textsuperscript{4}

\textsuperscript{1}California Institute of Technology, \textsuperscript{2}CERN, \textsuperscript{3}Grotto Networking, \textsuperscript{4}Tongji University, \textsuperscript{5}University of Southern California, \textsuperscript{6}Yale University

July 20, 2017, IETF99, Prague
Takeaway from IETF98

• ALTO can provide information on different resources to improve the performance of dataset transfers and data analytics applications.
  – In data center networks of the Compact Muon Solenoid (CMS) experiment, network resources are not always the bottleneck.
• ExaO: a multi-resource orchestrator for CMS applications.
Update in IETF 99

- Expand the application scenario
  - Previous: resource orchestration for science applications (ExaO).
  - Current: a unified resource orchestration framework for geo-distributed, multi-domain data analytics (Unicorn).

- Describe the Unicorn framework
  - Add resource view extractor, workflow converter, resource demand estimator, entity locator, etc. into the framework.
  - Add the detailed workflow for WG review.

- Restructure the document
  - Update abstract and discussion sections.
  - Add an example to show how ALTO can reveal fine-grained data locality information.
  - Describe how resource view extractor works.
Multi-Domain, Geo-Distributed Data Analytics

- **Vision**: Different organizations contribute various resources, e.g., sensing, computation, storage and networking resources, to collaboratively collect, share and analyze extremely large amounts of data.

  - Example: the CMS experiment, coalitions between different organizations, cloud exchange, etc.
Multi-Domain, Geo-Distributed Data Analytics

• **Goals**: production deployments of a new class of intelligent, software-defined global systems which
  – achieves efficient utilization of a large set of distributively-owned, heterogeneous resources;
  – maintains the autonomy and privacy of resource owners.

• **Solution**: a unified resource orchestration framework
  – An architecture for general multi-domain, geo-distributed data analytics
Why ALTO?

• Existing systems (HTCondor, Hadoop, YARN, Mesos, etc.) only provide coarse-grained information on resources, leading to inefficient resource allocation decisions.

• ALTO provides on-demand fine-grained information on different resources to support optimal resource orchestration.
Example

- Job J needs dataset X as input.
- Data center A and B each has a copy of X and can place J in the same rack as X is stored.
- Hadoop:
  - Resource information: $dist_A(J, X) = dist_B(J, X) = 2$
  - Job placement: execute J either at site A or site B
- ALTO:
  - Resource information:
    - $dist_A(J, X) = dist_B(J, X) = 2$
    - $bw_A(J, X) = 100Mb/s, bw_B(J, X) = 1Gb/s$
  - Optimal job placement: execute J at site B
Unified Resource Orchestration (Unicorn)

• **Resource supply**: use ALTO to provide the view of computing, storage and networking resources from different sites
  – Expand the capability of *abstract network element* (ANE) to provide an abstract view of resources.

• **Resource demand**: a set of tools for automatic, effective resource *demand estimation* for data analytics jobs

• **Resource orchestration**: use the views from ALTO for deep site orchestration among virtualized clusters, storage subsystems and subnets to successfully co-schedule CPU, storage and networks.
Unicorn: Architecture

- Workflow Converter
- Resource Demand Estimator
- Multi-Resource Orchestrator
- Manage resource demand dynamic
- Manage resource supply dynamic
- Matching demand and supply

- Resource View Extractor
- Entity Locator
- ALTO Client
- ALTO Server

- Entity Locator
- Resource View Extractor
- ALTO Client
- ALTO Server

- Execution Agents
Related ALTO extensions

- **ALTO Unified Property (adopted as a WG document)**
  - Retrieve properties of entities (e.g., endpoint, ane, etc.) in the cluster

- **ALTO Path Vector (adopted as a WG document)**
  - Retrieve the properties of a set of ane's shared by a set of data analytics flows

- **ALTO Cost Calendar (adopted as a WG document)**
  - Retrieve time-dependent endpoint cost

- **ALTO Routing State Abstraction**
  - Compress the information retrieved by ALTO path vector into a minimal, equivalent view

- **ALTO Flow Cost Service**
  - Retrieve cost information of flows instead of src-dst endpoint pair
Resource View Extractor (RVE)

- Previously called ANE aggregator.
- The ALTO client collects various information about different entities from different ALTO services.
- Current design: RVE works as an independent module instead of an ALTO service.
- It first assembles such information to form a raw resource view.
  - This view may have redundancy.
- It then uses a lightweight algorithm proposed in ALTO-RSA to compress the raw view into a minimal, equivalent view and pass to the orchestrator.
Design Issue: Scalability

- One data analytics job may consist of many low-level tasks. Tasks may have precedence relationships between each other.
- Querying the resource view for each task would cause huge overhead.
- Solution approach: selectively sampling
  - Tasks are often repeated or similar.
  - In one job, only some tasks will become the bottleneck.
Importance to ALTO WG

• Unicorn provides a template architecture for single-domain/multi-domain data center resource optimization, a major use case of ALTO listed in the WG Charter.

• In addition to RFC7285, Unicorn applies several ALTO extensions (WG documents: cost calendar, path vector and unified property map, etc.) to collect resource information from different sites.

• As an informational document, it will provide key insights and experience in the deployment use of ALTO services in a very large and public data analytics project.
Next Steps

• **Draft**
  – Continue to document the design and experience of Unicorn.
  – Add specific examples of using different ALTO services in the Unicorn framework.
  – etc.

• **Milestones**
  – Pre-production deployment of Unicorn by IETF 100.
  – Production deployment by IETF 102-103.