ALTO Integration and Implementation
Supporting CERN Data Management
(FTS/Rucio Integration)

Presenters: Jordi Ros Giralt, Mario Lassnig, Mihai Patrascoiu, Y. Richard Yang
On behalf of team (Jensen Zhang, Kai Gao, Lauren Delwiche, Mahdi, Ryan Yang)

March 27, 2023

IETF 116
Overview: LHCONE

- Part of CERN infrastructure (experimental facilities, LHCOPN, LHCONE)
- Topology: 600 distributed storage systems, distributed globally (170 data centers, in 127 sites, across 40 countries)
- Workload: support data movement for four LHC experiments, and also Belle II, Pierre Auger Observatory, NOvA, XENON, and JUNO
- Traffic: 2022, the aggregated outgoing traffic just from CERN to its ten largest connected data centres: 457 Petabytes of data.
Overview: Related Software Stack

- Data-Intensive Workflows
- Data Management / Transfer Orchestration (e.g., Rucio)
- Data Transfer Scheduling (e.g., FTS)
- Transfer Data Plane (e.g., GridFTP, XRootD, HTTP)
- Internet Transport Layer (e.g., TCP, TCP/Cubic, BBR)
- Networking Layer (e.g., traditional networking, AutoGOLE/SENSE, NOTED, Programmable net)

Our focus

High-level goals:
- Efficiency
- Fairness
Outline

- Overview
- Transport scheduling integration (ALTO/FTS)
Transfer Scheduling (FTS): Objective and Design

Related objectives:
- Efficiency control: Avoid overloading transfer resources (both network and storage), fully utilize all capacity
- Fairness/allocation: resource allocation beyond congestion

Mechanisms:
- Keeps transfer queue for each src/dst pair (pipe)
- Adjusts # concurrent TCP connections per pipe
- Dispatches transfer if allowed by concurrency level

New but not fully integrated mechanisms:
- Each file transfer (src->dst) is marked as on behalf of an activity of an experiment

Data-Intensive Workflows

Data Management / Transfer Orchestration (e.g., Rucio)

Transfer Scheduling (e.g., FTS)

Transfer Data Plane (e.g., GridFTP, XRootD, HTTP)

Internet Transport Layer (e.g., TCP, TCP/Cubic, BBR)

Networking Layer (e.g., traditional networking, AutoGOLE/SENSE, NOTED, Programmable net)
ALTO/FTS Objective: Application-Defined Networking

**Diverse, High Level Resource Models**

**ALTO/FTS Scheduler**

- Universally available
- Fast, efficient, robust building block
- But single resource allocation (fairness) model

**Diverse Network Settings**
ALTO/FTS, also called TCN: Light-weight control plane on top of fully distributed TCC
Resource Model:
Project 1 pipes: Pipe 1, Pipe 2
R1: <Project 1, Plink 1> <= 5G
R2: <Project 1, Plink 2> <= 10G
R3: <Project 1, Plink 3> <= 10G

App Provided State:
Pipe1.traffic = 3G, Pipe2.traffic = 9G

ALTO Provided State:
Pipe 1: {Plink 1, Plink 3, Plink 4}.
Pipe 2: {Plink 2, Plink 3, Plink 5}.

Resource use by project on physical resources:
Plink1.traffic = 3G, Plink2.traffic = 9G,
Plink3.traffic = Pipe1.traffic + Pipe2.traffic = 12G

Penalty for resource control constraints:
P(R1) = 0 (Plink1 = 3G <= 5G) , P(R2) = 0 (Plink2 = 9G <= 10G)
P(R3) = 2 (Plink3 = 12G > 10G)
ALTO/FTS Visibility

IETF ALTO Protocol as Front End

Other Sources
- GeoIP

Control Plane
- Config->FIB
- adjac
- subnets
- BGP input

Data Plane Control: FIB
- Looking Glass
- P4

Data Plane Data: Sample
- G2 snapshot
  - Samples (netflow, sflow)
  - PerfSonar (ICMP)
  - <switch interface, pkt attr>
  - <src, dst, <metric, val>>

Equivalent Class
- Other Sources
- Control Plane
ALTO/FTS First Hop Visibility

Query Example (ECS with path vector extension)

```
cat request-cern.json
{
  "cost-type": {
    "cost-metric": "one-path",
    "cost-mode": "array"
  },
  "endpoint-flows": [ 
    {"sps": [ipv4:137.138.8.101],
    "dps": [ipv4:134.158.84.23, ipv4:144.16.112.112] }
  },
  "sps": [ipv4:192.16.106.254],
  "dps": [ipv4:140.115.32.101] 
},
"one-property-names": [ "next_hop", "as_path" ]
}
```

Routing Plane Retrieval (Looking Glass of CERN and GEANT)

Implementation

Jensen/Kai/Lauren
ALTO/FTS Control Implementation

Integration into FTS 3.12

- Extend database schema for pipes (t_link_config) to support resource control specification (tcn_abs_limit, tcn_rel_weight)
- Implement ALTO/TCN Optimizer class for ALTO/TCN control loop
- Implementing ZeroOrder Gradient with Integral, Quadratic Distance function
- Add new optimizer mode (kOptimizerAggregated) to enable ALTO/TCN optimizer
Outline

• Overview
• Transport scheduling integration (ALTO/FTS)
• Transport orchestration integration (ALTO/Rucio)
ALTO/Rucio Objective: Uniform Orchestration Selection

Rucio Orchestrator

Uniform ALTO Sorting Interface

Diverse Network Visibility Resources
ALTO/Rucio Using Query Expression

Step 1: Configuration
Configure ALTO client at Rucio server to fetch visibility using ALTO

Step 2: Express Sorting using ALTO/Rucio Expression
ALTO sorting expression enables Rucio download command to sort replicas based on a combination of distances and properties, e.g.,

```
BY=as_hopcount,delay_ow WHERE continent="EU"
```

Map properties of ANEs into end-to-end metrics
ALTO/Rucio Using Query Expression: Default/Backup GeoIP/Distance

- Providing geoip property using the standard ALTO endpoint property service [RFC 9240]
- Providing geo distance between endpoints using the standard ALTO Endpoint Cost Service (ECS) [RFC 7285]
Summary: Current ALTO/FTS+Rucio: 3 Main Components

- Visibility of Multidomain Infrastructure (e2e aggregation and select control assets)
- Scheduling of Data Movement Tasks (build on FTS Integration)
- Data Flow Orchestration (build on Rucio)
Status and Next Steps

- Implementations
  - ALTO/FTS
    - Visibility: looking glass first-hop links (e.g., CERN border links to peers)
    - Control: Zero-order stochastic gradient algorithm, event-driven programming
    - Scale: 200x40, targeting 600x600 full mesh
    - Resource model: Full linear model
  - ALTO/Rucio
    - Fully integrated, uniform interface

- Deployment
  - Target full production workload in summer 2023, for HL-LHC Data Challenge
Backup Slides
ALTO/FTS Control Details

- Integral, quadratic distance function
- Zero-order stochastic rounding

\[ U(\tau) = \left( \sum_{i=1}^{K} w_i \tau_i \right) - \eta \cdot d(\tau, t \cdot K)^2 \]
Basic ALTO/FTS Benchmarking ⇒ Real Topology (ESnet)

1. **7.39x** total BW utilization.
2. **5.54x** Max RCT improvement. (Short-tailed)

Global Objective, Zero-order gradients, and Resource Control Constraints.

Setting: 30 <src, dst> pipes, one request per pipe, each request 20K transfers, file size = 100MB, the total in the workload is 60TB.
Resource Control goal: all equal weights.
## Basic Benchmarking: Results

**FTS-SG vs. ALTO/TCN FTS**

**BW utilization and resource control constraints satisfaction**

<table>
<thead>
<tr>
<th></th>
<th>FTS Semi-Gradient (FTS-SG)</th>
<th>ALTO/TCN FTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total BW Utilization</strong></td>
<td>Not optimal (6GB unused BW).</td>
<td>Fully utilized BW.</td>
</tr>
<tr>
<td><strong>Resource Control</strong></td>
<td>BW shares not satisfied.</td>
<td>BW allocated close to 2:1 spec.</td>
</tr>
<tr>
<td><strong>RCT</strong></td>
<td>Max RCT = 20,000 slots.</td>
<td>Max RCT = 4,000 slots.</td>
</tr>
</tbody>
</table>

*FTS-SG depends on correct configuration (e.g., high enough default). ALTO/TCN is fully automated.*

---

### FTS-SG Output

2 x 20TB Pipes on a Single Link

- **Throughput (Gbps):**
  - Request 1: 0, 2, 4, 6, 8, 10, 12
  - Request 2: 0, 2, 4, 6, 8, 10, 12

- **BW lim:**
  - 2 GB

- **Time Slot:**
  - 0 to 10000

### ALTO/TCN FTS

2 x 20TB Pipes on a Single Link

- **Throughput (Gbps):**
  - Request 1: 0, 2, 4, 6, 8, 10, 12
  - Request 2: 0, 2, 4, 6, 8, 10, 12

- **BW lim:**
  - 2 GB

- **Time Slot:**
  - 0 to 4000
Request Performance Distribution

1. 2.03x mean RCT improvement.
2. 2.49x max RCT improvement.

Global Objective, Zero-order gradients, and Resource Control Constraints.

Setting: Similar to previous slide, but with modification to include more requests to show more details: 100 <src, dst> pipes, one request per pipe, each request 5K transfers, file size = 100MB.

Resource Control goal: all equal weights.
From All-Arrival Workload ⇒ Dynamic Arrival Workload

Per flow comparison of ALTO and FTS Request Completion Times
50 request with incremental arrival time on ESnet

8.0x improvement in RCT when using ALTO/TCN.
(Global Objective, Full Zero Order)

Setting: ESnet (67 nodes), selected 50 active pipes; each pipe has transfer workload arrives according to a arrival distribution (Poisson arrival, with parameter 1200 (every 200 time slot); Each replication request has N(40k, 20k) files, file size is 100MB.
2.97x improvement in convergence speed (# transient slots) with using first-order gradients.

First-order integrated ALTO/TCN vs Full Zero-order ALTO/TCN

Setting: ALTO/TCNology ESnet, selected 10 pipes, infinite backlog.
A Semi Zero-Order Gradient Alg Optimizing for Each Pipe

Keep track of the exponential moving average (EMA) of throughput.

\[ E_i(t + 1) = \alpha T_i(t + 1) + (1 - \alpha)E_i(t) \]

Update the number of connections based on EMA:

\[
\begin{cases} 
  n_i(t + 1) = n_i(t) + 1 & E_i(t + 1) \geq E_i(t); \text{ Lines 9,11} \\
  n_i(t) - 1 & RL_B(E_i(t + 1)) < RL_B(E_i(t)); \text{ Line 4} \\
  n_i(t) & \text{else}
\end{cases}
\]

Theorem: In a Throughput-Deterioration Model, semi zero order will achieve throughput that is \( \leq \frac{1}{\sqrt{B}} \) of the optimal (under default settings).