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

draft-xiang-alto-multidomain-analytics-02

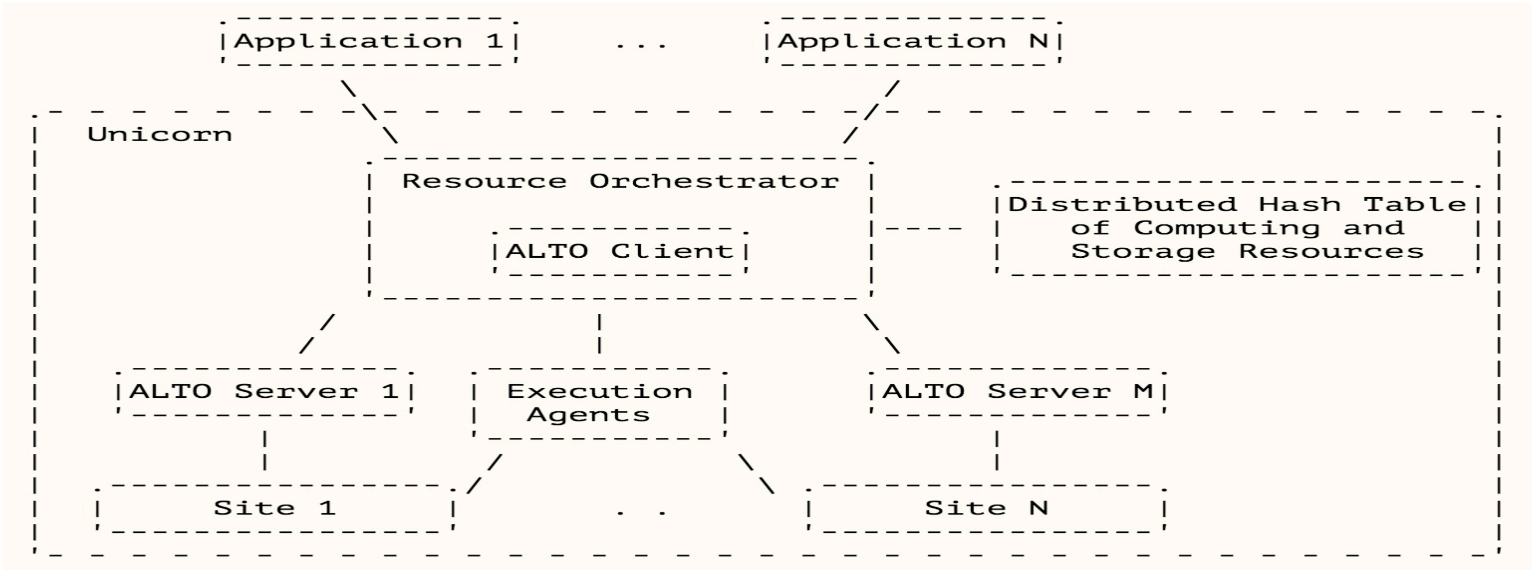
Qiao Xiang[1,2], Franck Le[3], Y. Richard Yang[1,2],
Harvey Newman[4], Haizhou Du[1], J. Jensen Zhang[1]

[1] Tongji University, [2] Yale University,
[3] IBM Watson Research Center,
[4] California Institute of Technology

July 16, 2018, IETF 102 ALTO

Takeaway from IETF 101

- Substantial updates for document review:



Three-phase resource discovery

1. ALTO EPS to discover locations and properties of computing and storage resources;
2. ALTO ECS to discover the connectivity between computing and storage resources
3. ALTO PV extension to discover the networking resource sharing between flows.
   - Propose an ALTO extension to support accurate, privacy-preserving resource discovery across multiple domains.

Update for IETF 102

- Two technical updates for the resource abstraction discovery phase (Phase 3).
  - Update the design of the privacy-preserving multi-domain resource abstraction aggregation protocol .
    - The new design does not require a chaining aggregation process between different ASes.
  - Introduce a super-set projection technique to improve the scalability.

Phase 3: Resource Abstraction Discovery

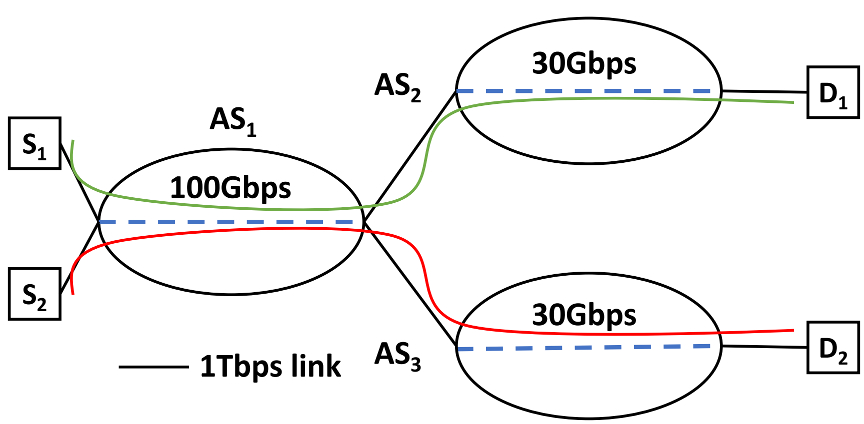
- **Previous design**: an ALTO-extension for privacy-preserving multi-domain resource information aggregation, which returns the intersected capacity region of all networks in ALTO PV extension.

```
                    .-----------------------.
                    | Resource Orchestrator |
                    |                       |
                    |     .-----------.     |
                    |     |ALTO Client|     |
                    |     '-----------'     |
                    '-----------------------'<
Encryption Key and /                          \ Ciphered MCP
Multipart Cost    /                            \ Response
Property Queries v   Key, Ciphertext            \
         .--------.  and MCP   .--------.         .--------.
         |ALTO    |  Queries   |ALTO    |         |ALTO    |
         |Server 1|  ------->  |Server 2|-- ... -->|Server N|
         '--------'            '--------'         '--------'
```

Representation of capacity region after the aggregation:

$$
\begin{aligned}
69x_1 + 61x_2 + 11x^s_{11} + 58x^s_{21} + 50x^s_{31} &= 4340,\\
71x_1 + 118x_2 + 49x^s_{11} + 22x^s_{21} + 69x^s_{31} &= 7630,\\
170x_1 + 184x_2 + 95x^s_{11} + 75x^s_{21} + 89x^s_{31} &= 14420,\\
59x_1 + 129x_2 + 34x^s_{11} + 25x^s_{21} + 95x^s_{31} &= 7000,
\end{aligned}
$$

- Example:

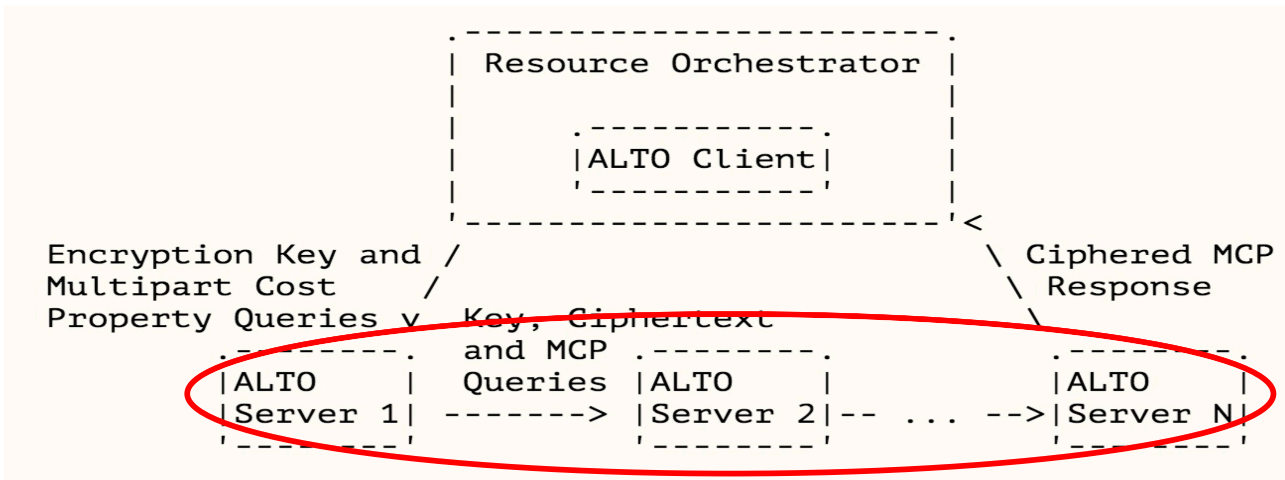


$$
\begin{aligned}
&\Pi_1(F_1) : \{x_1 + x_2 \le 100\}\\
&\Pi_2(F_2) : \{x_1 \le 30\}\\
&\Pi_3(F_3) : \{x_2 \le 30\}.
\end{aligned}
$$

Representation of capacity region before the aggregation:

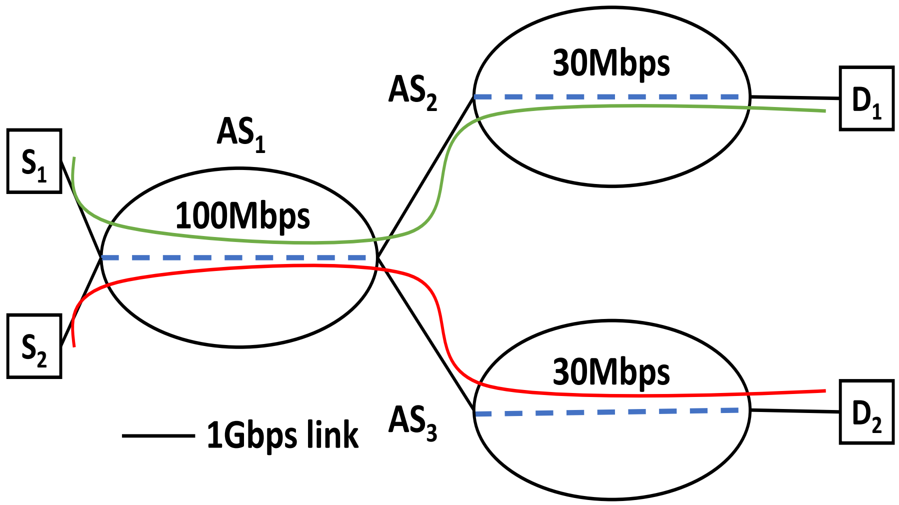
Phase 3: Resource Abstraction Discovery

- **Issue**: The aggregation process is a chaining process across all ALTO servers, which will take a long time for the ALTO client to get the final result.



- In the -02 draft, we design a new privacy-preserving aggregation service, which does not require the chaining aggregation process.
- **Basic idea**: Each ALTO server disguises its own set of linear inequalities in the ALTO-PV response with an obfuscating algorithm we developed.

New Privacy-Preserving Aggregation Service: An Example



Representation of capacity region before the aggregation:

$$\Pi_1(F_1) : \{x_1 + x_2 \leq 100\}$$
$$\Pi_2(F_2) : \{x_1 \leq 30\}$$
$$\Pi_3(F_3) : \{x_2 \leq 30\}.$$

```
                    .-------------------------.
                    | Resource Orchestrator   |
                    |                         |
                    |     .------------.      |
                    |     |ALTO Client|       |
                    |     '------------'      |
                   >'-------------------------'<
                  /Disguised    /|\            \
Multipart Cost   / Response      |              \
Property Queries v               |               \
        .---------.          .---------.          .---------.
        |ALTO     |          |ALTO     |          |ALTO     |
        |Server 1| <------->|Server 2|<--...  -->|Server N|
        '---------' Limitedly'---------'          '---------'
                    Shared Secret
```

AS$_1$

$$\begin{aligned} 15x_1 + 14x_2 + 15x_3 + 4x_4 + 0x_5 &= 1130 \\ 53x_1 + 50x_2 + 53x_3 + 2x_4 + 0x_5 &= 4910 \\ 96x_1 + 97x_2 + 96x_3 + 4x_4 + 0x_5 &= 9540 \\ 38x_1 + 37x_2 + 38x_3 + 1x_4 + 0x_5 &= 3420 \end{aligned}$$

AS$_2$

$$\begin{aligned} 56x_1 + 0x_2 - 4x_3 + 57x_4 + 3x_5 &= 1740 \\ 22x_1 + 0x_2 - 4x_3 + 24x_4 + x_5 &= 680 \\ 78x_1 + 2x_2 - x_3 + 74x_4 + 3x_5 &= 2250 \\ 25x_1 + 0x_2 + 0x_3 + 25x_4 + 0x_5 &= 770 \end{aligned}$$

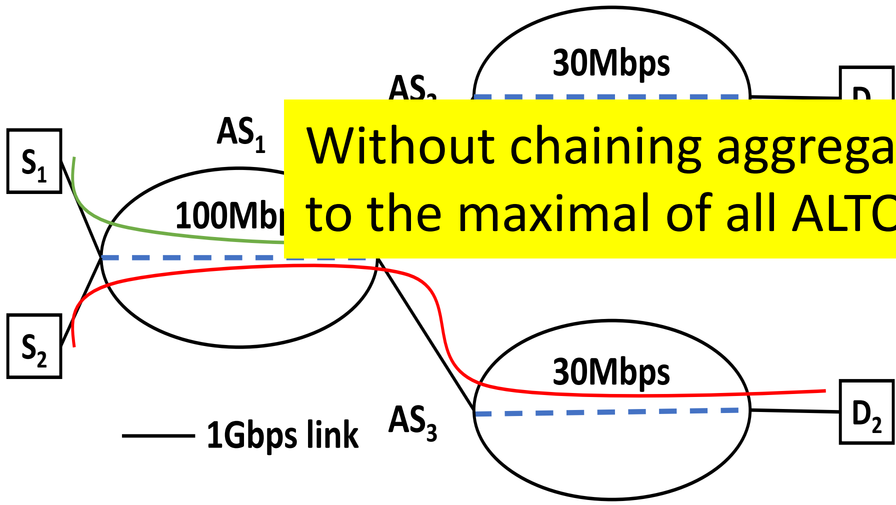
AS$_3$

$$\begin{aligned} -2x_1 + 47x_2 + 0x_3 - 3x_4 + 47x_5 &= 1470 \\ -4x_1 + 68x_2 + 0x_3 - 4x_4 + 68x_5 &= 2040 \\ -4x_1 + 85x_2 + 0x_3 - 3x_4 + 86x_5 &= 2630 \\ -4x_1 + 91x_2 + 0x_3 - 2x_4 + 94x_5 &= 2810 \end{aligned}$$

$$\boxed{\begin{aligned} 69x_1 + 61x_2 + 11x^s_{11} + 58x^s_{21} + 50x^s_{31} &= 4340, \\ 71x_1 + 118x_2 + 49x^s_{11} + 22x^s_{21} + 69x^s_{31} &= 7630, \\ 170x_1 + 184x_2 + 95x^s_{11} + 75x^s_{21} + 89x^s_{31} &= 14420, \\ 59x_1 + 129x_2 + 34x^s_{11} + 25x^s_{21} + 95x^s_{31} &= 7000, \end{aligned}}$$

Representation of capacity region after the aggregation:

New Privacy-Preserving Aggregation Service: An Example



Without chaining aggregation, the aggregation latency is reduced to the maximal of all ALTO server->client latencies .

Representation of capacity region before the aggregation:

```
                          .-------------------------.
                          | Resource Orchestrator   |
                          |                         |
                          |      .------------.     |
                          |      |ALTO Client|      |
                          |      '------------'     |
                         >'-------------------------'<
                         /Disguised    /|\          \
 Multipart Cost         /  Response     |            \
 Property Queries      v                |             \
            .---------.           .---------.          .---------.
            |ALTO     |           |ALTO     |          |ALTO     |
            |Server 1| <-------->|Server 2|<--... -->|Server N|
            '---------'  Limitedly'---------'          '---------'
                         Shared Secret
```

AS1

$$
\begin{aligned}
15x_1 + 14x_2 + 15x_3 + 4x_4 + 0x_5 &= 1130\\
53x_1 + 50x_2 + 53x_3 + 2x_4 + 0x_5 &= 4910\\
96x_1 + 97x_2 + 96x_3 + 4x_4 + 0x_5 &= 9540\\
38x_1 + 37x_2 + 38x_3 + 1x_4 + 0x_5 &= 3420
\end{aligned}
$$

AS2

$$
\begin{aligned}
56x_1 + 0x_2 - 4x_3 + 57x_4 + 3x_5 &= 1740\\
22x_1 + 0x_2 - 4x_3 + 24x_4 + x_5 &= 680\\
78x_1 + 2x_2 - x_3 + 74x_4 + 3x_5 &= 2250\\
25x_1 + 0x_2 + 0x_3 + 25x_4 + 0x_5 &= 770
\end{aligned}
$$

AS3

$$
\begin{aligned}
-2x_1 + 47x_2 + 0x_3 - 3x_4 + 47x_5 &= 1470\\
-4x_1 + 68x_2 + 0x_3 - 4x_4 + 68x_5 &= 2040\\
-4x_1 + 85x_2 + 0x_3 - 3x_4 + 86x_5 &= 2630\\
-4x_1 + 91x_2 + 0x_3 - 2x_4 + 94x_5 &= 2810
\end{aligned}
$$

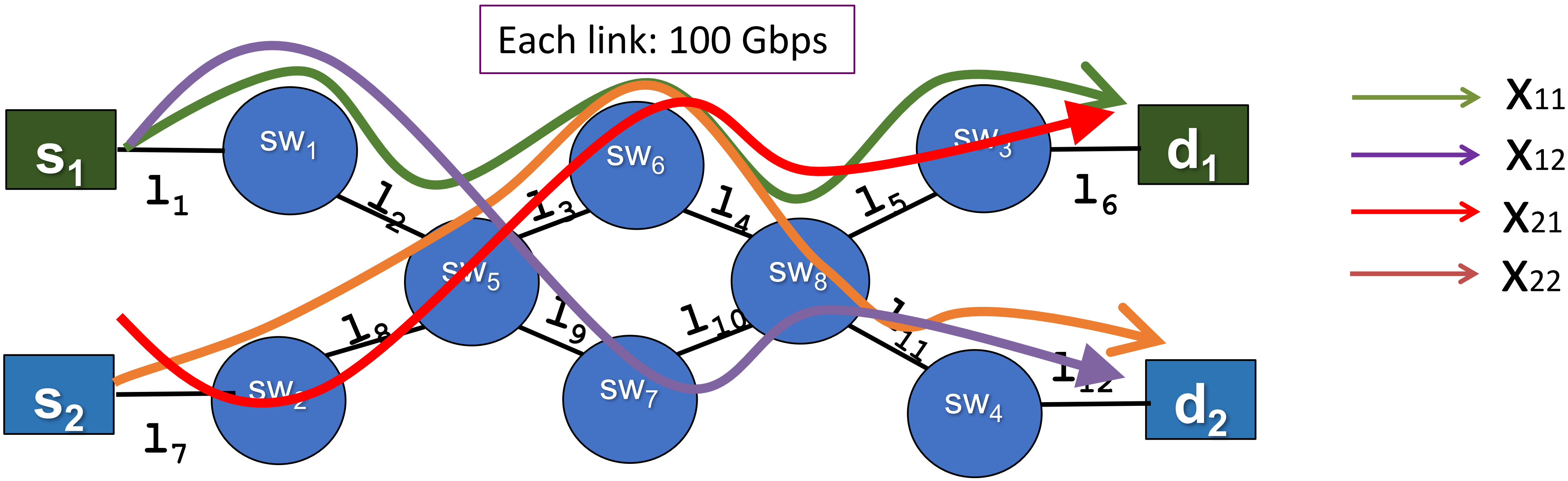
$$
\begin{aligned}
69x_1 + 61x_2 + 11x^s_{11} + 58x^s_{21} + 50x^s_{31} &= 4340,\\
71x_1 + 118x_2 + 49x^s_{11} + 22x^s_{21} + 69x^s_{31} &= 7630,\\
170x_1 + 184x_2 + 95x^s_{11} + 75x^s_{21} + 89x^s_{31} &= 14420,\\
59x_1 + 129x_2 + 34x^s_{11} + 25x^s_{21} + 95x^s_{31} &= 7000,
\end{aligned}
$$

Representation of capacity region after the aggregation:

Improve Scalability: Super-Set Projection

- **Issue**: In collaborative science experiments, the number of data analytics jobs is huge. Repeatedly querying ALTO servers and making ALTO servers compute the responses for every new job would raise scalability issue.
- **Current design proposal**: Super-set projection.
- **Basic idea**: For each ALTO server in each network, let it precompute the routing information, and the resource sharing information for a set of flows, whose source and destination is the combination of all ingresses and egresses of the networks.
  - E.g., Assume a network with M ingress and N egress, precompute the route and bandwidth sharing for a set of M*N flows.
  - When a new PV query comes in, the ALTO server can projects the pre-computed set of linear inequalities for the M*N flows based on the ingresses and egresses of the flows in the PV query, to get the resource sharing information for this query.

Super-Set Projection: Example



- Only two ingresses (l_1, l_7) and two egresses (l_6, l_{12})
- Pre-computed set of linear inequalities:

$x_{11} + x_{12} \leq 100$, for link $\{l_1, l_2\}$
$x_{11} + x_{21} \leq 100$, for link $\{l_5, l_6\}$
$x_{11} + x_{21} + x_{22} \leq 100$, for link $\{l_3, l_4\}$
$x_{21} + x_{22} \leq 100$, for link $\{l_7, l_8\}$
$x_{12} + x_{22} \leq 100$, for link $\{l_{11}, l_{12}\}$
$x_{12} \leq 100$, for link $\{l_9, l_{10}\}$

- When a PV query comes with two flows (s1, d1) and (s2, d2), the projected result is:

$x_{11} \leq 100$, for link $\{l_1, l_2, l_5, l_6\}$
$x_{11} + x_{22} \leq 100$, for link $\{l_3, l_4\}$
$x_{22} \leq 100$, for link $\{l_7, l_8, l_{11}, l_{12}\}$

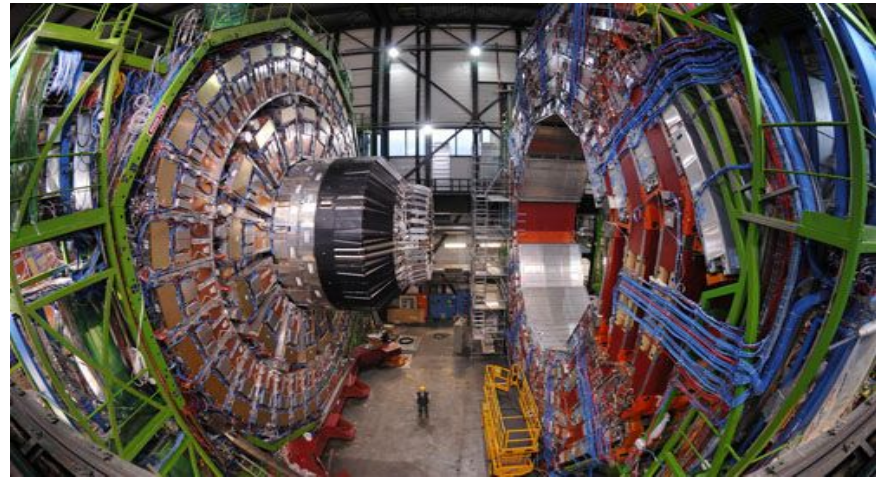
Next Steps

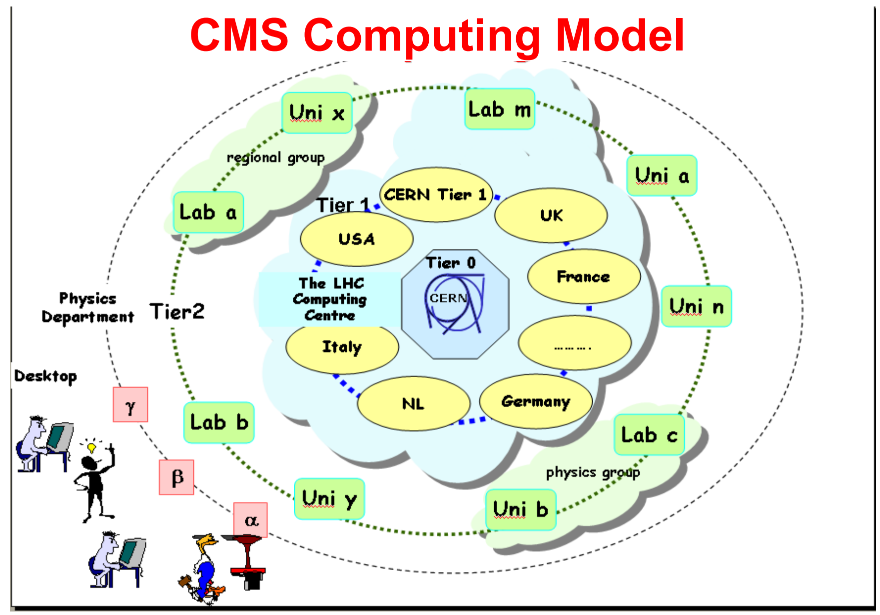
- Privacy-preserving information aggregation:
  - This is a first proposal to address the security/privacy concerns of using ALTO.
  - Interests in moving it to a formal extension (standard track)?
- Super-set projection
  - The current design focuses on resource abstraction discovery (phase 3).
  - Sending the pre-computed set of linear inequalities to the ALTO client, who can do projection by itself, could further reduce the discovery latency.
  - How to extend this design to the other two phases (endpoint and path discoveries) without raising additional privacy concerns?
- The overall system is under the final review phase of SuperComputing'18.

Backup slides

Recap: Multi-Domain, Geo-Distributed Data Analytics

- **Settings**: 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 in Large Hardon Collider.

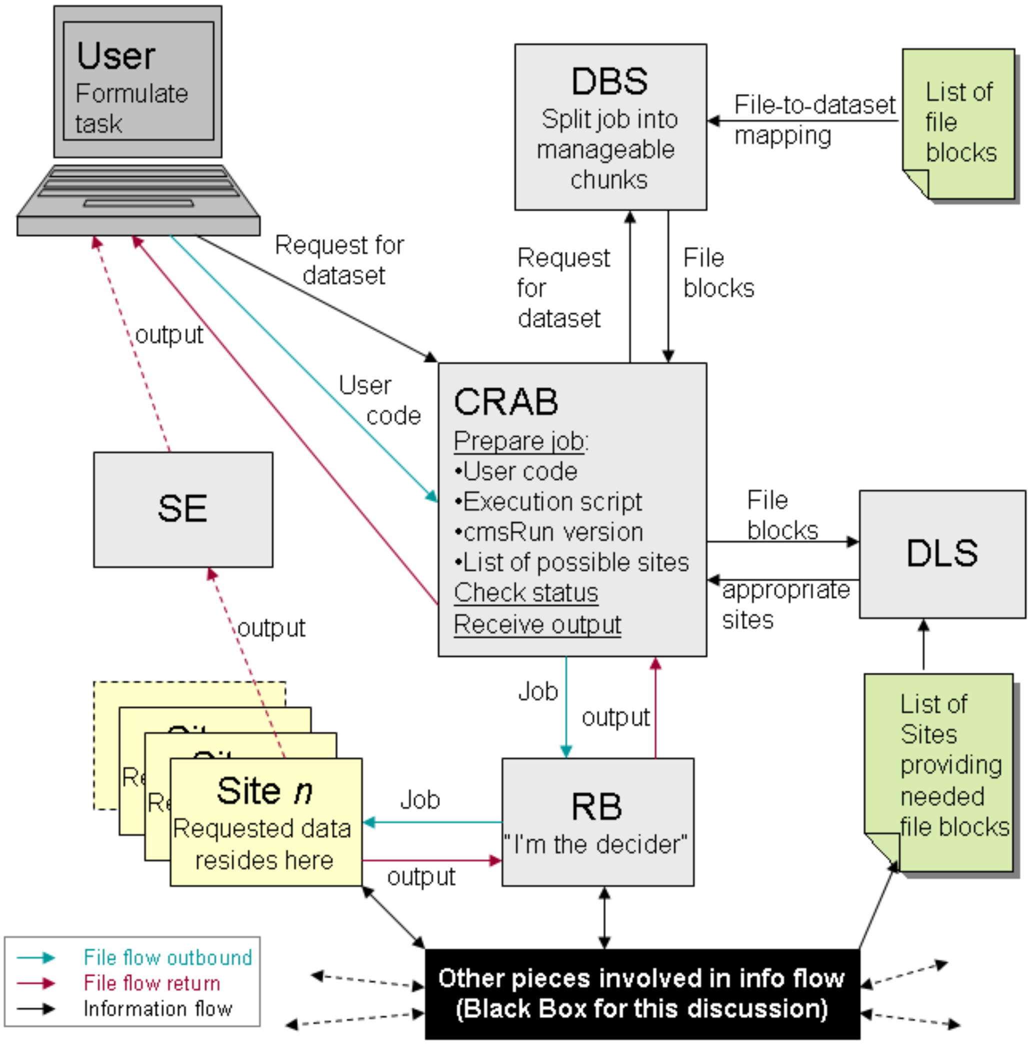




Current CMS Data Analytics Work Flow

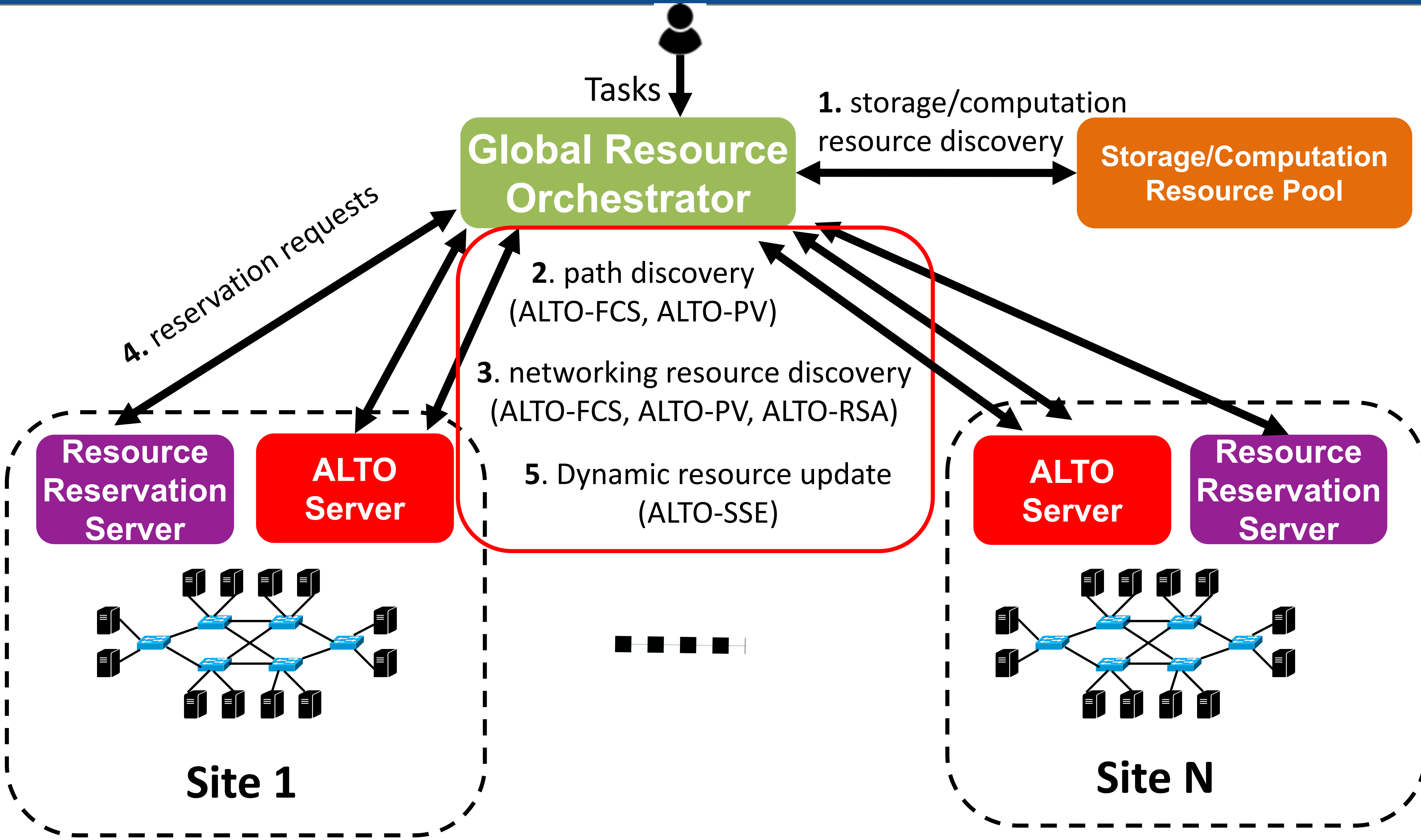
- **Factors determining data analytics task delay.**
  - Task decomposition (parallelization).
  - Data transmission from input dataset location to computation nodes.
  - Data transmission from computation nodes to output dataset sites.

- Current CMS workflow.
  - Simple, manual parallelization.
  - Opportunistic, network-unaware computation node assignment.
  - Opportunistic, network-unaware output stage out.



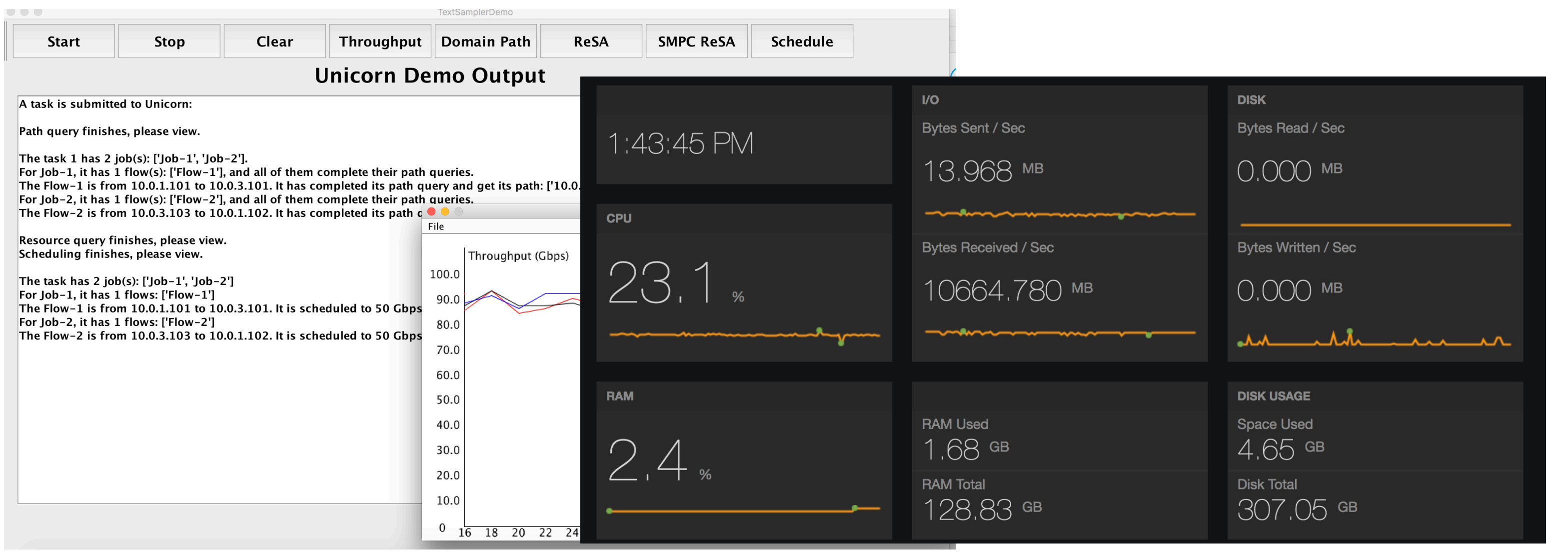
Source: https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookAnalysisWorkFlow

Architecture



Unicorn Implementation and Demonstration

- Orchestrator: ~2700 LoC Python code
- ALTO server: ~3000 LoC Java code
- Resource reservation server:
  - fast data transfer (FDT), FireQoS, OpenvSwitch, etc.
- Network controllers: OpenDaylight, Kytos
  - ONOS and Ryu are under development
- Demonstrated on different topologies at SuperComputing 2017 [2].



[2] Xiang, *et.al.*, "Unicorn: Unified Resource Orchestration for Multi-Domain, Geo-Distributed Data Analytics", in INDIS Workshop 2017.