Dirigo: Self-scaling Stateful Actors For Serverless Real-time Data Processing

Le Xu
University of Texas at Austin

Real-time Data Processing Frameworks

Serverful Environment (e.g., CaaS)
- Monolithic applications
- User involved in deployment constantly
- Single-tenant, serverful environment

Serverless Environment (e.g., FaaS)
- Application written in the format of function DAG
- Resources multiplexed finely among applications
- Multi-tenant, serverless environment

Real-time Data Processing Meets Serverless

Dirigo: Comparing to Existing Systems

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Serverful Streaming Platform</th>
<th>Serverless Architecture</th>
<th>Dirigo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Utilization</td>
<td>Low</td>
<td>High (multi-tenancy)</td>
<td>High (multi-tenancy)</td>
</tr>
<tr>
<td>Provisioning Granularity</td>
<td>Coarse Grained (per worker)</td>
<td>Fine Grained (per message)</td>
<td>Fine Grained (per message)</td>
</tr>
<tr>
<td>Provisioning Scope</td>
<td>Application scope</td>
<td>Platform scope</td>
<td>Platform scope</td>
</tr>
<tr>
<td>Native State Management</td>
<td>✓</td>
<td>✓</td>
<td>✓ (dual mode actor)</td>
</tr>
<tr>
<td>Performance-Driven</td>
<td>✓</td>
<td>✓</td>
<td>✓ (scheduling architecture with plugged in policies)</td>
</tr>
</tbody>
</table>

Dirigo’s Contributions
- Fine-grained provisioning support
- Ease of deployment
- Processing semantics support

Dirigo’s control messages overhead

Dual-mode actor protocol in action

Our vision
Designing a multi-tenant, cloud service based on serverless architecture, that achieves both high resource utilization as well as performance isolation.

Ongoing Evaluation

Dirigo Architecture

Dirigo Policies

Dual-Mode Actors

Using Dual-mode Actor to Coordinate Snapshot on a Distributed Operator

Tasks and States

Our vision
Designing a multi-tenant, cloud service based on serverless architecture, that achieves both high resource utilization as well as performance isolation.