Henge: Intent-Driven Multi-Tenant Stream Processing

Faria Kalim, Le Xu, Sharanya Bathey, Richa Meherwal, Indranil Gupta

Distributed Protocols Research Group

Department of Computer Science
University of Illinois at Urbana Champaign

http://dprg.cs.uiuc.edu/
Henge allows stream processing jobs to satisfy user-specified performance requirements while reducing costs by performing online resource reconfigurations in a multi-tenant environment.
A Typical Deployment

Per-job clusters → overprovisioning
A Typical Deployment
Low level metrics e.g., queue sizes, CPU load as performance indicators

Job 1
Job 2
Job 3
Job 4
A Typical Deployment
Low level metrics e.g., queue sizes, CPU load as performance indicators
A Typical Deployment

Low level metrics e.g., queue sizes, CPU load as performance indicators

Manual scaling
Intent-Driven Multi-Tenancy
Intent-Driven Multi-Tenancy

Efficient resource usage across multiple users

→ Multi-tenancy
Intent-Driven Multi-Tenancy

Efficient resource usage across multiple users
➔ Multi-tenancy

Application-aware adaptation to user requirements
➔ Intent-driven Multi-tenancy
Intent-Driven Multi-Tenancy

Efficient resource usage across multiple users
→ Multi-tenancy

Application-aware adaptation to user requirements
→ Intent-driven Multi-tenancy

<table>
<thead>
<tr>
<th>Job Description</th>
<th>Service Level Objective (SLO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finding ride price</td>
<td>Latency &lt; 5 s</td>
</tr>
<tr>
<td>Analyzing earnings over time</td>
<td>Throughput &gt; 10K/hr.</td>
</tr>
</tbody>
</table>
Problem

How can we achieve user-facing service level objectives for stream processing jobs on multi-tenant clusters?
Problem

How can we achieve user-facing service level objectives for stream processing jobs on multi-tenant clusters?

- Latency
- Throughput
Absolute Throughput SLOs are not Useful
Absolute Throughput SLOs are not Useful
Absolute Throughput SLOs are not Useful

[Graph showing rate (Tuples/s) over Day 1 and Day 2 with SLOs indicated by horizontal lines.]

Rate (Tuples/s)
Absolute Throughput SLOs are not Useful
Absolute Throughput SLOs are not Useful

Workload Variability

Rate (Tuples/s)

SDSC Input  SDSC Output  EPA Input  EPA Output

Input  Output

Day 1  Day 2

SLO?
Absolute Throughput SLOs are not Useful
Absolute Throughput SLOs are not Useful

**Juice**: fraction* of the input data processed by the job per unit time.
Jobs benefit even below SLO threshold

Job Utility Functions
Jobs benefit even below SLO threshold

Job Utility Functions

Utility function for a single job
Jobs benefit even below SLO threshold

Job Utility Functions

Henge’s goal → Maximize the total utility of the cluster

Utility function for a single job
Background: Stream Processing Topologies (Jobs)

Logical DAG for a Word Count Job
Diamond Topology

Star Topology
Background: Stream Processing Jobs

So it goes…

[“So”]

[“it”]

[“goes”]

[“So”]

[“goes”]

...
Background: Stream Processing Jobs

```
[“So”]
[“it”]
[“goes”]
```

```
[“So it goes…”]
```

```
[“So”]
[“goes”]
[“it”]
```

Parallelism → 2

Executors (Threads)
Background: A Physical Deployment
Background: A Physical Deployment

Spout  Splitter
Count  Count

Workers
Henge’s Cluster-Wide State Machine

Not Converged

Converged

Total System Utility < Total Expected Utility
Henge’s Cluster-Wide State Machine

Reconfiguration

Not Converged

Total System Utility < Total Expected Utility

Converged

Reversion or Reconfiguration

Reduction
Reconfiguration

De-congest operator by increasing parallelism level of executors

1) Reconfiguration

2) Reconfiguration
Reconfiguration

De-congest operator by increasing parallelism level of executors

3) **Black-list** topologies that show less than $\Delta\%$ improvement

1) Reconfiguration

2) Reconfiguration
Bottlenecks

- Spout
- Splitter
- Count
- Count

Workers
Bottlenecks

Reconfigurations

Spout  Splitter
Splitter  Splitter
Count  Count

Workers
Bottlenecks

Reconfigs.

High
Load

Spout
Splitter
Count
Splitter
Splitter
Count
Workers

SLO-Satisfying Job

- Spout
- Splitter
- Count
- Worker configurations

Load

- High
- Satisfies SLA
Bottlenecks

Reconfigs.

High Load

Reduction
Bottlenecks

Reduction

Reconfigs.

High Load

Reduction
Reduction

Reconfigurations $\rightarrow$ drop in utility
Reduction

Reconfigurations $\rightarrow$ drop in utility
If high CPU load on majority of machines, reduce parallelism for operators that
   a) are in topologies that satisfy their SLO
   b) are not congested
Reversion

Reconfigurations $\rightarrow$ drop in utility and reduction is not possible. **Revert** to a past configuration that provided best utility.
Evaluation

Real-world workloads:
  Yahoo!
  Twitter
  Web log traces
Experimental Setup:
  10-40 node Emulab cluster
Reducing cost and achieving high utilities

![Graph showing SLO Satisfaction (%) vs Percentage of Resources Used (%)]

- **Henge**
- **Default Storm**
Reducing cost and achieving high utilities

93.5% utility at 40% resources
Reducing cost and achieving high utilities

93.5% utility at 40% resources

100% utility at 60% resources

SLO Satisfaction (%)

Percentage of Resources Used (%)
Adapting to a Diurnal Pattern
Fewer reconfigurations are required once a job has adjusted to max load.
Can Henge do better than manual configuration?
Can Henge do better than manual configuration?

Henge does better in the 15th to 45th percentile, and is comparable later.
Scaling Cluster Size

The graph shows the average number of reconfigurations for different sizes of clusters. As the size of the cluster increases, the average number of reconfigurations remains relatively constant.
Scaling Cluster Size

Limited resources entail more reconfigurations to reach max. utility

Average Reconfigurations

Size of Cluster

10 20 30 40
More Results

Henge can:

- handle dynamic workloads
  - abrupt e.g., spikes & natural fluctuations
  - gradual e.g., diurnal patterns
- satisfy hybrid SLOs
- scale with number of jobs & cluster size
- gracefully handle failures
Summary

• Henge allows users to specify **performance intents** for their jobs
• Henge’s goal is to **maximize cluster-wide utility**
• The scheduler performs fine-grained **reconfigurations** to allow stream processing jobs to meet user-specified intents