SCIP under the hood: distributed computing with Dask

Phase 1
Lazy task graph construction using Dask datastructures

[input/1.tiff, input/2.tiff, ...]

Bag[(input/1.tiff), (input/2.tiff), ...]

loading

Bag[(input/1.tiff), (input/2.tiff), ...]

masking

Bag[(input/1.tiff), (input/2.tiff), ...]

filtering

Bag[(input/1.tiff), (input/2.tiff), ...]

feature extraction

DataFrame

<table>
<thead>
<tr>
<th>feature</th>
<th>SCM</th>
<th>stage</th>
<th>memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>feat1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>feat2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>feat3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

/output/features.parquet

Phase 2
Distributed task graph execution

DASK

Task Scheduler

Smart task scheduling
Fault tolerance
Load balancing
Memory management

Workers
execute tasks