Design Space Exploration: HyperMapper + Spatial

Artur Souza
Ph.D. student, UFMG
Client-Server Mode

• Allows software to call HyperMapper on demand
  • HyperMapper acts as an “optimization server”
  • Software acts as a client
Client-Server Mode

- Allows software to call HyperMapper on demand
  - HyperMapper acts as an “optimization server”
  - Software acts as a client
- Software launches HyperMapper as a separate process
  - HyperMapper becomes an internal step in the software
Client-Server Mode

• Allows software to call HyperMapper on demand
  • HyperMapper acts as an “optimization server”
  • Software acts as a client
• Software launches HyperMapper as a separate process
  • HyperMapper becomes an internal step in the software
• Client-server mode is triggered on the JSON:

```
“hypermapper_mode”: {
  “mode”: “client-server”
}
```
Client-Server Mode

- HyperMapper no longer calls the black-box function internally
Client-Server Mode

• HyperMapper no longer calls the black-box function internally
• Both processes run independently and communicate to coordinate the search
  • HyperMapper chooses configurations to explore
  • Client evaluates configurations and returns performance
Client-Server Mode

- HyperMapper no longer calls the black-box function internally
- Both processes run independently and communicate to coordinate the search
  - HyperMapper chooses configurations to explore
  - Client evaluates configurations and returns performance
- Software can be written in any language
  - E.g. Spatial is written in Scala
Client-Server Protocol

- Client and server communicate via a csv-like protocol:

Client (software) → scenario.json → Server (HyperMapper)
Client-Server Protocol

- Client and server communicate via a csv-like protocol:

<table>
<thead>
<tr>
<th>Request 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, x_2$</td>
</tr>
<tr>
<td>9, 0</td>
</tr>
<tr>
<td>9, 2</td>
</tr>
<tr>
<td>5, 1</td>
</tr>
</tbody>
</table>

Client (software) ➔ Server (HyperMapper)
Client-Server Protocol

- Client and server communicate via a csv-like protocol:

\[ x_1, x_2, \text{value} \]
- 9,0,5.8
- 9,2,1.2
- 5,1,12.7
Client-Server Protocol

- Client and server communicate via a csv-like protocol:
Client-Server Protocol

• Client and server communicate via a csv-like protocol:

Client (software) \rightarrow \text{Server (HyperMapper)}

\begin{align*}
x_1, x_2, \text{value} \\
4, 1, 3
\end{align*}
Client and server communicate via a csv-like protocol:

**Request 1**

\[ x_1, x_2, 3, 2 \]
Client-Server Protocol

• Client and server communicate via a csv-like protocol:
Client-Server Protocol

- Client and server communicate via a csv-like protocol:

Constrained Optimization

• Some Spatial configurations are impossible to synthetize
  • We say these configurations are “infeasible”
Constrained Optimization

• Some Spatial configurations are impossible to synthetize
  • We say these configurations are “infeasible”
• Goal is to find the best feasible value
Constrained Optimization

• Some Spatial configurations are impossible to synthetize
  • We say these configurations are “infeasible”
• Goal is to find the best feasible value
• HyperMapper can handle these constraints
Constrained Optimization

• Some Spatial configurations are impossible to synthetize
  • We say these configurations are “infeasible”
• Goal is to find the best feasible value
• HyperMapper can handle these constraints
• Activated in the json:

```
"feasible_output": {
  "enable_feasible_predictor": true
},
```

• Hands-on demo in the DSE Advanced Topics section
Resources

• Client-server mode:
  • https://github.com/luinardi/hypermapper/wiki/Client-Server-Mode

• Constrained optimization:
  • https://github.com/luinardi/hypermapper/wiki/Chakong-Haimes

• Advanced DSE topics after the break:
  • Multi-objective optimization
  • Constrained optimization
  • Prior-injection