Comparison of REST and GraphQL interfaces for OPC UA

Riku Ala-Laurinaho
8.4.2021
Motivation

- Easy access for shop floor data
- Bring web technologies to industrial domain
- Faster and easier application development

Fig. 1. Industrial overhead crane located at Aalto Industrial Internet Campus.

Fig. 2. Communication between client and OPC UA server. Redrawn from [1].
A (very) brief introduction to OPC UA

- Standardized protocol for industrial communication developed by OPC UA foundation
- Platform-independent, works with almost any devices
- Supports client-server and publish-subscribe models

& REST

- De facto architectural style in Web
- Commonly used with APIs to provide machine-readable data
- Client requests data from server over HTTP
- GraphQL is a query language and runtime
- Only the needed data is fetched and the response is predictable
- Multiple resources can be requested with a single request
- Hietala implemented GraphQL wrapper for OPC UA in his master’s thesis
## GraphQL vs. REST

<table>
<thead>
<tr>
<th></th>
<th>REST</th>
<th>GraphQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication model</td>
<td>Client-server</td>
<td>Client-server and subscriptions</td>
</tr>
<tr>
<td>Protocol</td>
<td>HTTP</td>
<td>HTTP and, for ex. WebSockets</td>
</tr>
<tr>
<td>Cache</td>
<td>At every layer</td>
<td>Application-specific</td>
</tr>
<tr>
<td>Scalability</td>
<td>Good</td>
<td>Medium</td>
</tr>
<tr>
<td>Interface</td>
<td>Uniform</td>
<td>Application-specific</td>
</tr>
<tr>
<td>Ease of use</td>
<td>Medium</td>
<td>Good (GraphiQL tool)</td>
</tr>
<tr>
<td>Ease of development</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>Bandwidth usage</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Performance</td>
<td>Medium</td>
<td>Good</td>
</tr>
</tbody>
</table>
Measurement setup

- Interfaces are middlewares on top of OPC UA
- GraphQL implementation by Hietala et al. [4] available at: https://github.com/AaltoIIC/OPC-UA-GraphQL-Wrapper
- REST interface by Cavalieri et al. [5] available at: https://github.com/OPCUAUniCT/OPCUAWebPlatformUniCT
- Python Free OPC UA library: https://github.com/FreeOpcUa/python-opcua

Fig. 5. Measurement setup for request execution times.
Request execution times

- OPC UA is significantly faster than REST and GraphQL
- REST is slightly faster than GraphQL with single values
- GraphQL outperforms REST with multiple values
TCP payloads (bandwidth use)

- OPC UA has very low TCP payloads
  - But no authentication was used
- GraphQL has smaller bandwidth usage than REST
  - Especially with multiple values because only one request is needed

Fig. 8. TCP payloads when reading 1 value.

Fig. 9. TCP payloads when reading 50 values.
Conclusion

- It was shown that REST/GraphQL interface for OPC UA server increases request execution times and bandwidth usage considerably.
- However, both improve the interoperability and makes OPC UA server more accessible.
- GraphQL offered better performance, lower bandwidth usage and is considered easier to use than REST.
- It is suggested that GraphQL interface is provided along with OPC UA server to combine the best sides of both.
References


Thank you!