Automatic RESTful Service Discovery for Web Connected Environments

Performance Evaluation of the Indexing Schema Construction

In this appendix, we evaluate the performance of constructing the indexing schema, IdS, which relates the existing static resources to their provided functions defined in F of FG. To this end, we conducted several experiments using different functions and resources graphs setups, and evaluated the IdS construction in terms of response time (ms) and memory usage (kb).

The IdS construction consists of:

- Retrieving the set of functions defined in FG with their corresponding signature (i.e., fsignature) based on the functions dependencies
- Traversing the existing resource graph containing the static resources to get their provided functions with their related resources (rsignature)
- Linking each function to the set of the static resources realizing it

Figure 1 and **Figure 2** illustrate respectively the response time and the memory usage of the tests conducted while varying the number of functions defined in FG. The number of the static resource in these tests is 2000. As shown in both figures, both the response time and the memory usage increase with the evolution of the number of functions. This is due to the calculations required to get the necessary signature of each function, and to link each function to the set of static resource matching it.

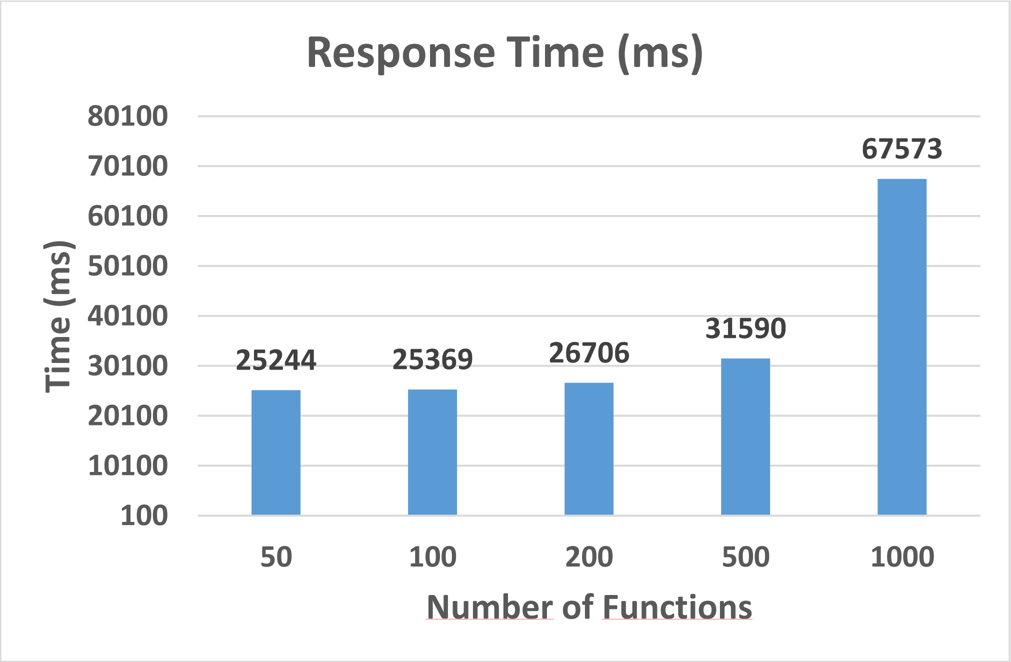

Figure 1- The response time of the tests conducted while varying the number of functions

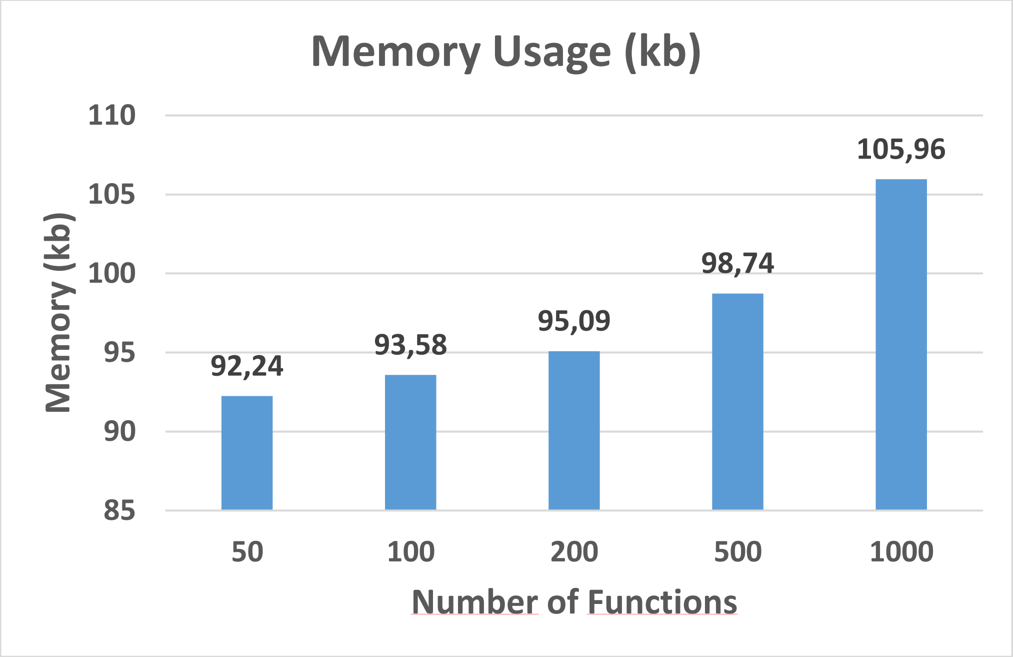

Figure 2- The memory usage of the tests conducted while varying the number of functions

Figure 3 and **Figure 4** show respectively the response time and the memory usage of the tests conducted while varying the number of static resources. In these tests the number of functions is 500. As it is seen in both figures, the response time and the memory usage increase when augmenting then number of resources. This is explained by the graph resource traversal to get the provided function of each resource with its related resources, and to the linking of each function to the set of resources answering it.

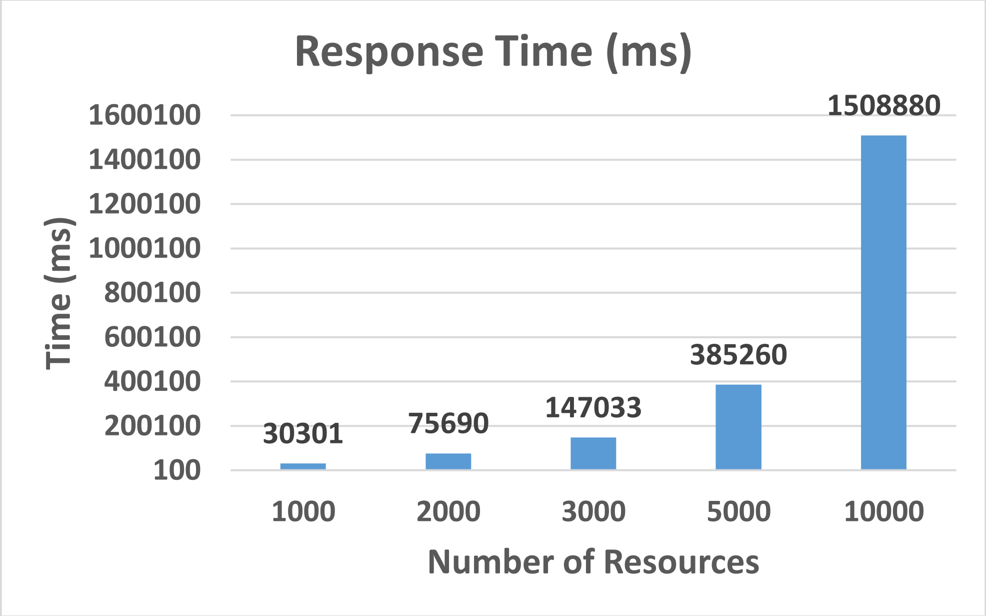

Figure 3- The response time of the tests conducted while varying the number of resources

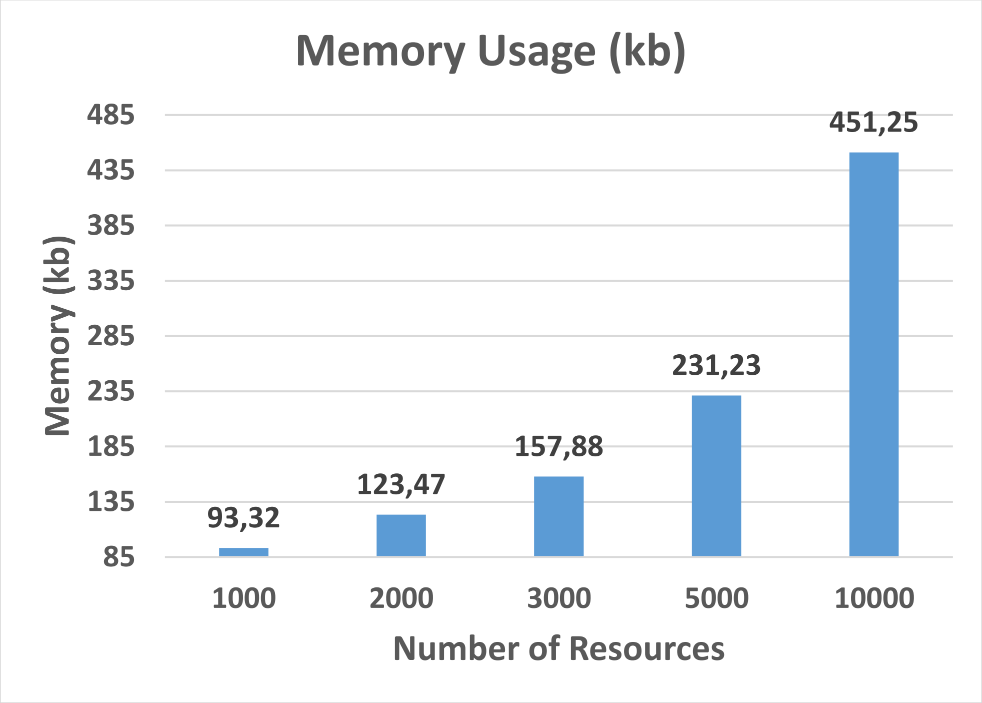

Figure 4- The memory usage of the tests conducted while varying the number of resources

From our experiments, we can see that the increasing number of functions and resources affects the construction of IdS in terms of response time and memory usage. However, as shown in the conducted tests, the evolution of these two aspects augments more with the growth of the number of resources.

In the current work, the defined IdS is generated every time there has been a change within the function graph (i.e., add/remove functions, change in the functions dependencies, etc.) or the resource graph (i.e., connect/disconnect static resources, add/remove functions provided by a static resource, etc.). In a later phase, we seek to update IdS dynamically.