Automatic K-Resources Discovery for Hybrid 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 linked static resources to their provided functions defined in the function graph, 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 (i.e., 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 1000. As shown in both figures, 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.

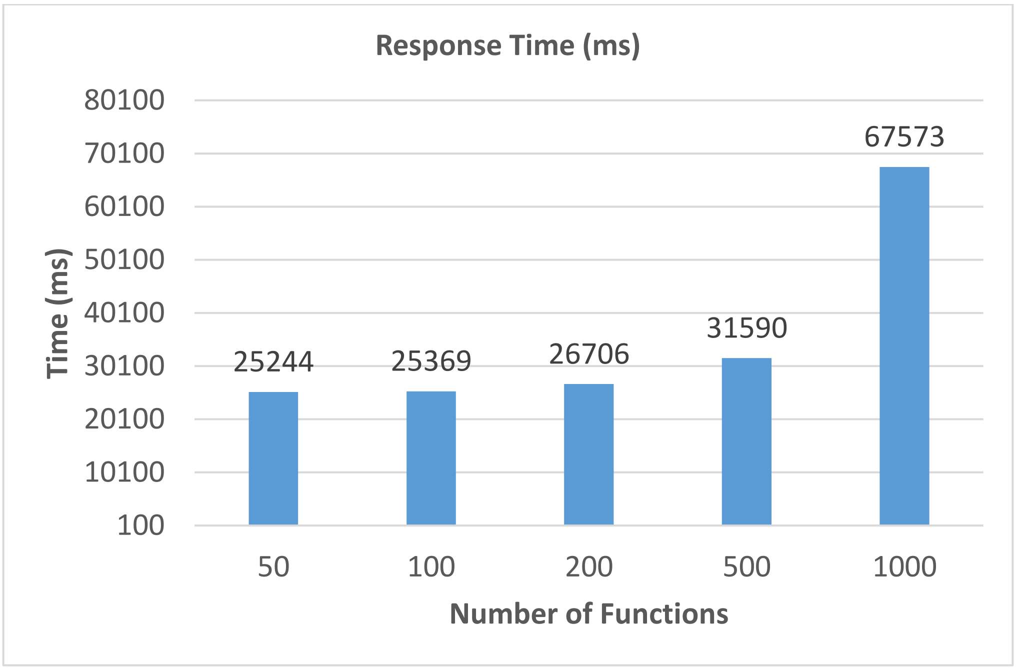

Figure 1- The indexing time of the tests conducted while varying the number of functions

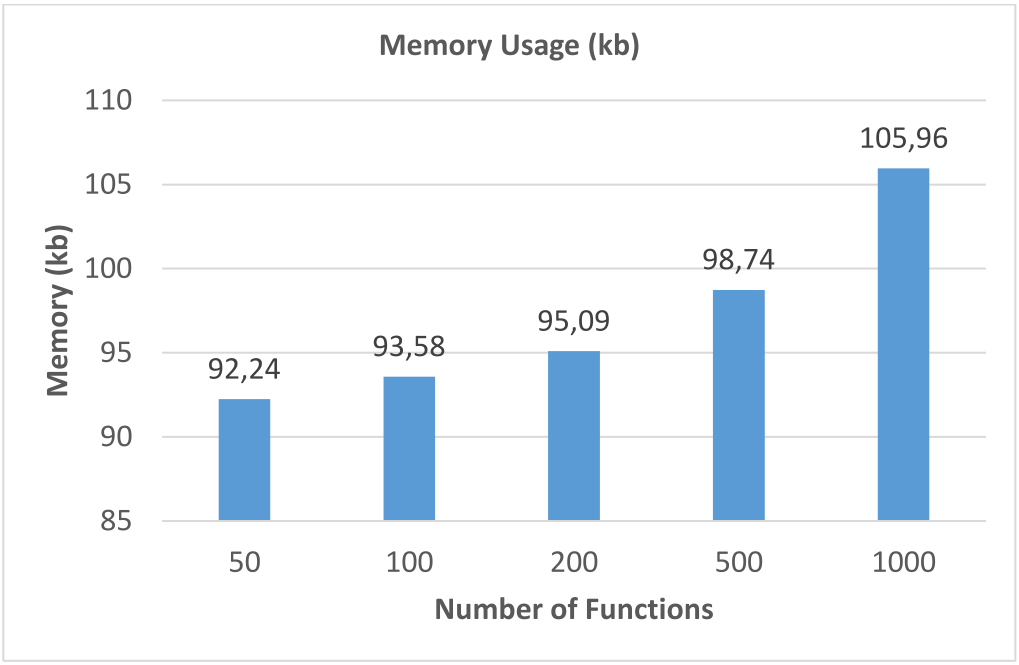

Figure 2- The indexing memory usage of the tests conducted while varying the number of functions

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 the 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.

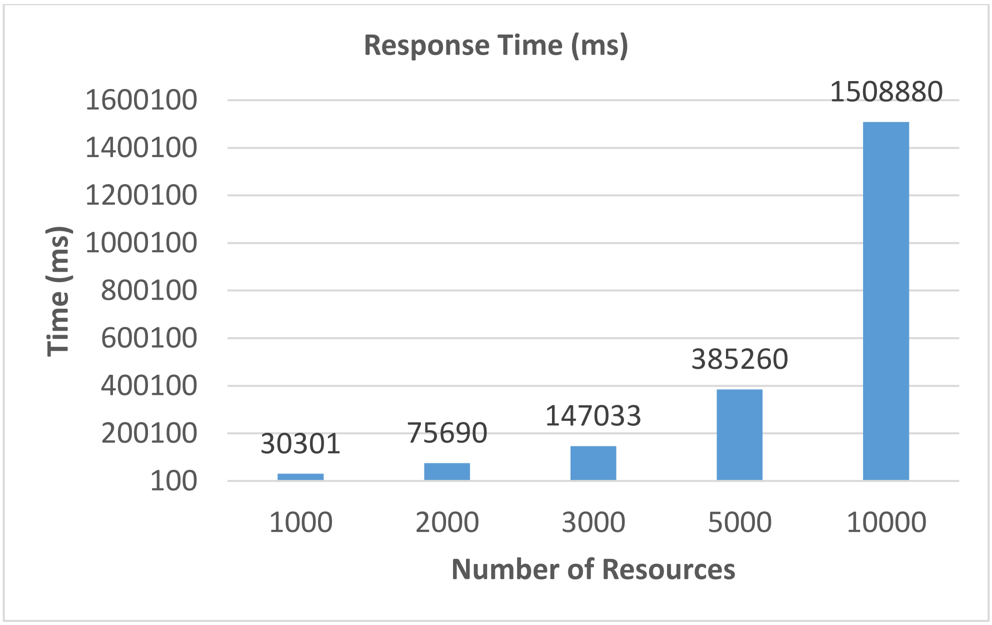

Figure 3- The indexing time of the tests conducted while varying the number of resources

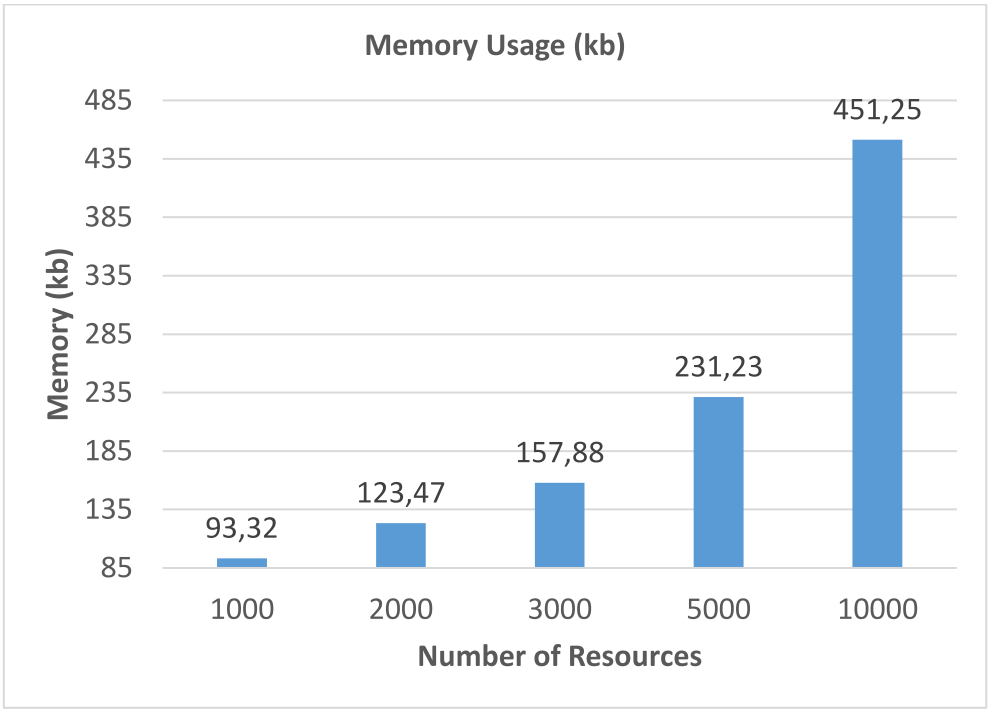

Figure 4- The indexing memory usage of the tests conducted while varying the number of resources

The experiments show that increasing the number of functions and resources affects the construction of the indexing schema in terms of response time and memory usage. However, these two aspects increase more with the growth of the resources number comparing to the functions number. In our work, the indexing schema is generated every time there is a change within the function graph (i.e., add/remove functions and change in the functions dependencies) or the resource graph (i.e., connect/disconnect static resources). Nevertheless, the indexing time shown in **Figure 1** has an exponential curve with the increase number of functions. The same graph pace appears in **Figure 3** and **Figure 4** related to the indexing time and indexing memory usage respectively, while increasing the resources number. These curves show that updating the indexing schema by regenerating it from the start requires lot of time and memory space when both functions number and resources number are relatively high. Although in real Web-based environments the number of resources does not normally exceed 10000, nor even the provided function are 1000, we seek in future works to improve the indexing schema performance construction by updating it dynamically without regenerating it from scratch every time, and thus, avoiding huge response time and lot of memory space.