LATE DECOMPOSITION OF APPLICATIONS

This briefing reports scientific evidence on how decompose quickly an application into services using MDE.

FINDINGS

- The findings presented in this briefing consider success as the application memory distribution among the software engineer defined services, with a little or no interference of software engineer in the distribution process.
- The project presents the successfully Apache Tomcat application and memory distribution among some defined services, although having an increase of memory usage when more simultaneous accesses were made in the decomposed application.
- We had two goals for this experiment: to determine if the approach can be used in practice in a real software and to investigate the impact of distribution in terms of performance and memory usage.
- There were five test cases: TC1: HTTP requests submitted to Tomcat's default management applications; TC2: same requests as TC1, but with 100 simultaneous clients; TC3: deployment of a web application called Hipergate into Apache Tomcat. TC4: HTTP requests made to the Hipergate application. These requests would be submitted by 10 simultaneous clients, three times, with a delay of 1 second between each time; and TC5: same requests as TC4, but with 100 simultaneous clients.
- There were four models used to distribute the application into services. DM1: this corresponds to the original Tomcat system in a single server; DM2: Tomcat was distributed between two servers with two services; DM3: Tomcat in four servers with four services. DM4: Tomcat in ten servers with ten services.
- All the four models were tested using all the five test cases.
- TC1 presents some interesting observations. By comparing DM1 with DM2, we observe a real memory separation occurring between the servers. In DM2, a big part of memory consumption still remains in the main Tomcat classes (represented by the first service), thus reducing memory consumption in individual servers during most part of the execution.
- In TC2, the distribution of memory consumption is not as clear as in TC1, probably because each individual server had to deal with many more simultaneous requests and remote method calls, and also instantiate more remote objects, which consume more memory than local objects.
- In TC3 more memory is used, when compared with TC1 and TC2. Although we can see memory sharing, again demonstrating that distribution is occurring correctly.
- In TC4 we observed that in distribution models DM2, DM3 and DM4 there is less memory consumption in individual servers than DM1. This is similar than TC1, although here there is not a higher dispersion in DM3.
- In TC5 we observed the same behavior of TC2. Because there are more simultaneous clients, each individual server has more memory consumption because of the higher number of remote objects being used.

Keywords:

MDE
Distributed architectures

Who is this briefing for?

Software engineering practitioners who want to make decisions about application architectures based on scientific evidence.

Where the findings come from?

All findings of this briefing were extracted from the late decomposition project conducted by Esperança et al.

To access other evidence briefings on software engineering:

http://www.lia.ufc.br/~cbsoft2017/en/