SOABench

Performance Evaluation of Service-oriented Middleware Made Easy

Domenico Bianculli • Walter Binder
Università della Svizzera italiana
University of Lugano - Switzerland

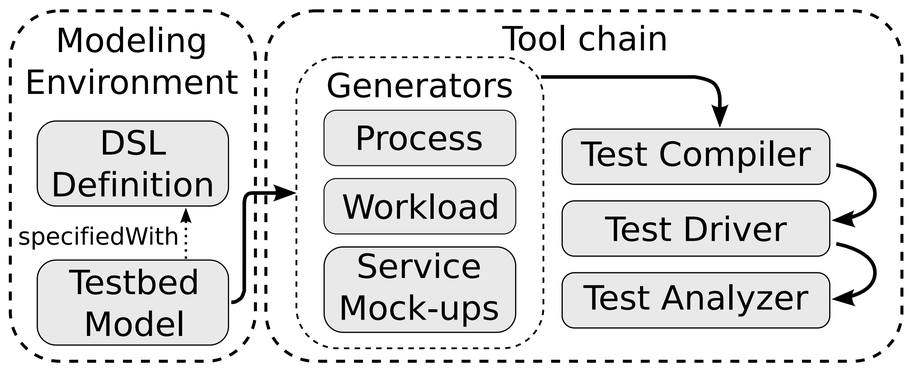
Mauro Luigi Drago
Politecnico di Milano - Italy

The problem

- Service-oriented architectures are realized by deploying and executing services on the top of a service-oriented middleware
- How to assess the performance of middleware components?
- Without automation, performance evaluation is cumbersome and error-prone

Our solution

The SOABench framework



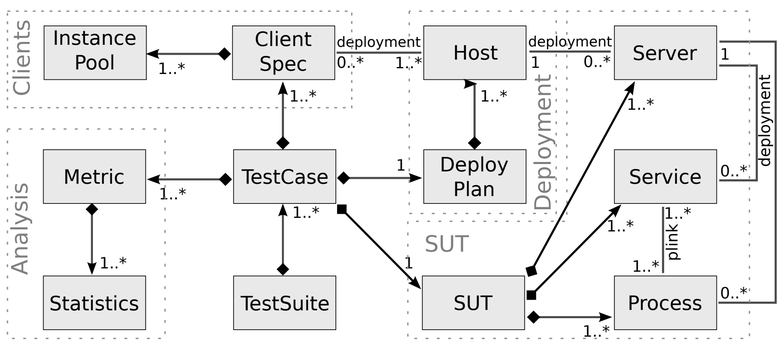
Features

- ✓ Technology independence
- ✓ Model-driven approach
- ✓ Repeatability
- ✓ Run-time evolution of testbeds
- ✓ Ability to process data and produce reports

Writing the testbed model

step 1

Meta-model for experimenting with SOAs



used to write the testbed model

with the testbed specification language (a DSL)

```
testsuite BpelBenchmarking {
library DefaultLib {
  serverSkeletons {
    skeleton AVOS5 {
      type = BPEL_ENGINE;
    }
    skeleton AXIS2 {
      type = SERVICE_ENGINE;
    }
  }
  component {
    host BpelServerHost {
      address = "bpelsvr.bpeltestbed.org";
      credentials {
        username = "bpelserver";
        type = SSH;
      }
      server BpelServer {
        location = "/home/AVOS5";
        relativeUrl = ":8080/avos5/procs/";
        skeleton = AVOS5;
      }
    }
  }
  serviceSkeletons {
    skeleton QosReplayService {}
  }
  service QSR_0R {
    deploymentServer = AxisServer_Axis2;
    skeleton = QosReplayService;
    mockSpec {
      type = invocations;
      instances = 50000;
      failureProb = 0.0;
      responseTime = Const(0.1);
    }
  }
}

testCases {
   testCase sequential_1{
     process sequential_proc_1 {
       deploymentServer = BpelServer;
       plink qosReplayServicePL-1 {
         type = fixed;
         port = "qosreplayservice-port";
         usedService = QSR_0R;
       }
     }
     instancePool pool_1 {
       instance {
         frequency = 0.4;
         data = "First Message";
       }
       instance {
         frequency = 0.6;
         data = "Second Message";
       }
     }
     clientSpec closed_clients_1 {
       number = 50;
       maxInvocations = 100;
       deploymentHosts = ClientHosts;
       wkld {
         type = closed;
         thinkTime = N(0.5,0.5);
       }
     }
   }
 }
}
```

Automated tasks

step 2

Generation of:
- mock-ups of the services to be used
- service compositions (BPEL processes)
- workload

step 3

Generation of:
- service deployment descriptors
- starting/stopping scripts

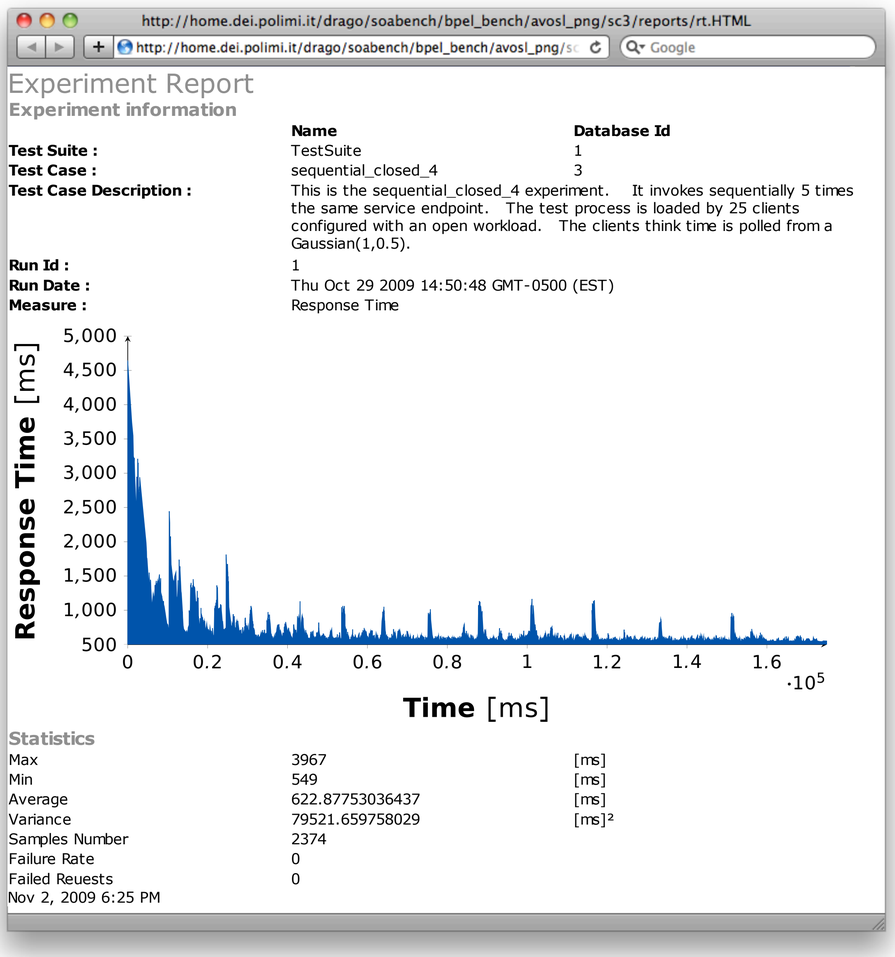
step 4

- Compilation of the experiment model, targeting a certain execution platform
- Execution of the experiments

step 5

- Collecting and processing run-time data
- Production of reports

Analysis



http://code.google.com/p/soabench/

Reference
D. Bianculli, W. Binder and M. L. Drago.
Automated performance assessment for service-oriented middleware: a case study on BPEL engines.
In Proc. of WWW 2010, pp. 141-150, April 2010, ACM.