Proposal for Bachelor Thesis in Software Engineering

Base information

Joakim Olsson laggmonkei@gmail.com
Johan Liljegren datanizze@gmail.com
Title (preliminary): Comparison between current open source Enterprise Service Buses
Academic Advisor: Nino D. Fogelström
Start date: 17/02/12
Proposal ok by date: -
Presentation date: -
Thesis type: research/industrial

Background

Enterprise Service Buses (ESB) [1] are pivotal in software integration and are becoming extremely important in company software solutions and infrastructures, future and ancient alike [2]. As such it is very important to accurately and repeatedly measure the performance of integration solutions available. There are currently several open source options available [3] and the current body of knowledge regarding their different performance aspects is severely lacking.

There has been some measurements and benchmarks done by the producers of the ESBs however these measurements have always been in favor of the ESB produced by the company making the measurements. Impartial performance measurement are especially lacking, there are several reports made by the producers of some solutions [4, 5, 6, 7, 8]. There has been some reasearch done in the area [9] however the number of ESB's tested are quite limited when compared to the most up to date list of popular ESBs [3]. Most importantly they do not explicitly declare what version they are using in their tests which makes their results hard to reproduce. Judging by the dates on which their papaer was completed it is clear that all the ESBs that they use have received major upgrades which makes it even more important to produce new measurements.

The main purpose of this thesis is to increase the knowledge of differences between modern open source ESBs. This will hopefully lead to it being easier to further develop these software products and make it easier for end users to compare and evaluate based on its specific needs.

Aims and objectives

Compare several enterprise service buses in a controlled environment covering a multitude of tests.

- Measure Ease of use
- Measure CPU usage
- Measure memory usage
- Usability during setup
- Test five ESB (Mule, WSO2, Servicemix, OpenESB and Blackbird)

Research questions

- R1: How does current open source ESBs stack up against each other? CPU/RAM/DISK comparison.
- R2: How does current open source ESBs scale?
- R3: How accurate and up to date is our current knowledge?
- R4: What is the perceived usability and ease of use during testing and setup?

Expected outcomes

The outcome will consist of large amount of graphs compiled from measurements during tests. There will also be a table compiled from subjective measurements regarding perceived usability and ease of use during the different stages of testing. It is important that the test methods used will be transparently reported and easily reproducable making "second opinions" for test confirmation a trivial task. This will improve the thesis validity in the sense that all facts and data are available making partial/twisted outcomes harder.

Research Methodology

Literature review

Proposals for search items:

- ESB, Enterprise Service Bus
- Integration
- MuleESB, WSO2, Servicemix, OpenESB, Blackbird ESB
- Performance measurement

We will combine the above keywords into search strings which we will use on Google Scholar, IEEE Xplore, Elin and other relevant web archives. We will primarily look at abstracts and results as a first line filter in order to find relevant articles. Those articles found relevant will be further investigated and have its references checked for relevance. We will also be mindful of tests found during the review in order to populate our own test suite with interesting, broad and most importantly unbiased tests.

Post-mortem analysis design

We will conduct a post-mortem analysis based on data acquired from tests performed in a controlled environment. The data that will be collected will mainly be focused on CPU and RAM usage during different amounts of throughput. The measurements are subject too change depending on discoveries made during literature review.

References

[1] F. Menge, "Enterprise Service Bus." http://kanagwa.com/assets/21/Esb1.pdf, 2007.

[2] J. Fenner, "Enterprise Application Integration Techniques." http://www.cs.ucl.ac.uk/staff/ucacwxe/lectures/3C05-02-03/aswe21-essay.pdf, 2003.

[3] A. Mehta, "11 Of Best Opensource ESB Tools." http://www.toolsjournal.com/integrations-articles/item/224-11-of-best-opensource-esb-tools, 2011.

[4] A. Perera, "WSO2 ESB Performance Testing ." http://wso2.org/library/1721, 2007.

[5] "Understanding Integration From A "Needs-Based" Perspective - Mule vs. JBoss ESB." http://www.mulesoft.org/comparing-mule-vs-jboss-esb, 2010.

[6] "Understanding Integration From A "Needs-Based" Perspective - Mule vs. Open ESB / Glassfish ESB." http://www.mulesoft.org/comparing-mule-vs-open-esb-glassfish-esb, 2010.

[7] "Understanding Integration From A "Needs-Based" Perspective - Mule vs. ServiceMix / Fuse ESB." http://www.mulesoft.org/comparing-mule-vs-servicemix-fuse-esb, 2010.

[8] "Mule Performance Test Results." http://www.mulesoft.com/downloads/Whitepaper_perf_test_results.pdf, 2008.

[9] A. P. Sanjay P. Ahuja, "." http://www.scirp.org/Journal/PaperInformation.aspx?paperID=6768, 2011.