Red Hat® JBoss® BPM Suite Performance Benchmark

WHITE PAPER — EXECUTIVE OVERVIEW

CONTENTS

Conceptualizing the Performance Framework

Benchmark Test Strategy

High-Level Key Findings

Conclusion

About Vizuri

Investigating poor performance in Business Process Management (BPM) implementations is an arduous task. BPM frameworks can be notoriously difficult to assess, measure and accurately predict how they will scale under normal use and load. There is not one "blanket statement" that we can provide, which takes into account each customer's unique business requirements and specific implementation design. When we are asked about performance, it becomes one of those critical evaluation questions, in which we utter a frustrating reply of "it depends." Estimating performance is increasingly difficult when contemplating a general and flexible BPM framework such as the Red Hat JBoss BPM Suite ecosystem.

We believe that there is a need for a suitable way to alleviate this frustration and ambiguity, while providing organizations with a reusable framework that can be used for evaluation against any BPM framework or process server. We have developed a BPM Test Framework that establishes a foundation organizations can model and exercise many different types of scenarios in simulation through a set of reusable resources. The results from performance simulations can be uniformly evaluated by providing a more concrete picture of how the Red Hat JBoss BPM Suite performs, while accounting for as much of the end-user process design and implementation variability as feasible.

This **Executive Overview** defines the approach, terminology, and provides the simulation results at a high-level. The **Technical Supplement** provides technical reviewers with a deep dive into the methods and raw data, which can be used for independent verification and evaluation.

Conceptualizing the Performance Framework

In our experience performing BPM consulting, we have seen the definition of BPM change by customer and within each stakeholder group (e.g., business unit, IT, C-Level, etc.). To provide a baseline from which to measure our performance assessment of the Red Hat JBoss BPM Suite, we will outline our assumptions and factors that dictate the context of what we are assessing.

BPM DEFINITIONS

When people hear that a software platform implements "BPM" automation, they often have one of several preconceived notions of what that means to them:

- An oversight engine that loosely defines and manages a business process,
- An orchestration engine that manages low level interactions between systems, or
- A platform to manage the lifecycle of documents.

Any one of these interpretations is correct within their stakeholder context. While the Red Hat JBoss BPM Suite can be used to perform low-level orchestration and/or manage document lifecycles, this evaluation will associate the Red Hat JBoss BPM Suite with the first category.

Since a general BPM Framework can be used to automate just about any process, we needed to narrow evaluation criteria to simple, generic workflows that provide feedback on key attributes for performance evaluation. This criterion would be relevant to the performance evaluation of any general BPM implementation engine.

NATURE OF BPM

When considering the implementation of business process automation through the use of a BPM tool or platform, we have to keep in mind that it is a declarative approach. Processes represent the definition, structure and flow of linked tasks, but should not contain the task implementation. The majority of the task work should be conducted outside of the process definition and invoked (e.g., asynchronous messaging, web service, etc.). Processes should spend the overwhelming majority of their time "sleeping" while waiting on a signal or human task to complete. The state of these "sleeping" processes is stored in the database and, while idle, consumes no resources on the runtime environment until awakened.

ANALYZING BPM PERFORMANCE

Usually, when investigating BPM performance, we find there is a significant amount of misplaced logic that is operating within the system tasks. This logic should be designed to operate at a lower level and in an asynchronous manner. Using a layered architecture to business automation is a common design approach that provides a business centric view, which rests atop the services layer in an orchestration framework. One example of this approach is the use of Camel running on an integration platform such as Red Hat JBoss Fuse.

To Download the Complete Document Visit

http://engage.vizuri.com/jbossbpmperformance

Experience • Innovation • Results

vizuri.com | blog.vizuri.com