Edited by XU Product Architecture Knowledge Transfer

PROJECT NAME / SUBJECT: **Solid Rock**

SPONSOR/PROJECT INITIATOR: Harald Müller

PROGRAM/PROJECT LEAD: Dirk Marwinski

LEAD ARCHITECT:

DEVELOPMENT: ☒ SAPLabs, mainly in Walldorf and Sofia ☐ Partner/ISV _____

I. MARKET AND PRODUCT BACKGROUND

Planned product release date: SAP NetWeaver New York

Underlying SAP NetWeaver release: SAP NetWeaver New York

Used SAP NetWeaver stacks: ☐ ABAP ☒ J2EE

Target platform (OS, DB):
- All

Targeted market segments / sub-segments:
- All

Use cases targeted by the product:
- Increase the robustness of SAP NetWeaver by enabling fast session failover
- Decrease overall memory consumption

Strategic goals SAP wants to achieve with the product:

n/a

Mandatory product capabilities to address goals, use cases, and target market:

n/a

II. ARCHITECTURE

Main Architecture Concepts and Decisions

The main idea behind the Solid Rock architecture is to increase the robustness of the J2EE Engine by reducing the number of sessions per java virtual machine and by separating the active from the inactive sessions.

By reducing the number of session handled by a java virtual machine (VM), the impact or a crash of a particular VM is reduced. Crashes can happen due to OutOfMemoryErrors, java virtual machine bugs, or other issues. As an example, consider a system with a single VM which handles 500 sessions. Should that VM crash, all 500 sessions are lost (and the work associated with it). In contrast, by distributing the work across eight virtual machines, a VM crash would mean the loss of approximately 60 sessions.

Separating the active sessions from inactive sessions means to move the session context for inactive sessions out of the address space a particular VM process, so it is not affected by a VM crash. Considering the “think time” of users, assume that from the 60 sessions handled by a particular VM only 10% of the sessions are active and the remaining 90% of the sessions are stored in an address space outside the address space of the processing VM. In this case a crash would only affect 6 sessions instead of 60. Compared to the starting point above (500 sessions) this is a major improvement.

© 2006 - SAP AG

The architecture diagram below outlines how these concepts are implemented:

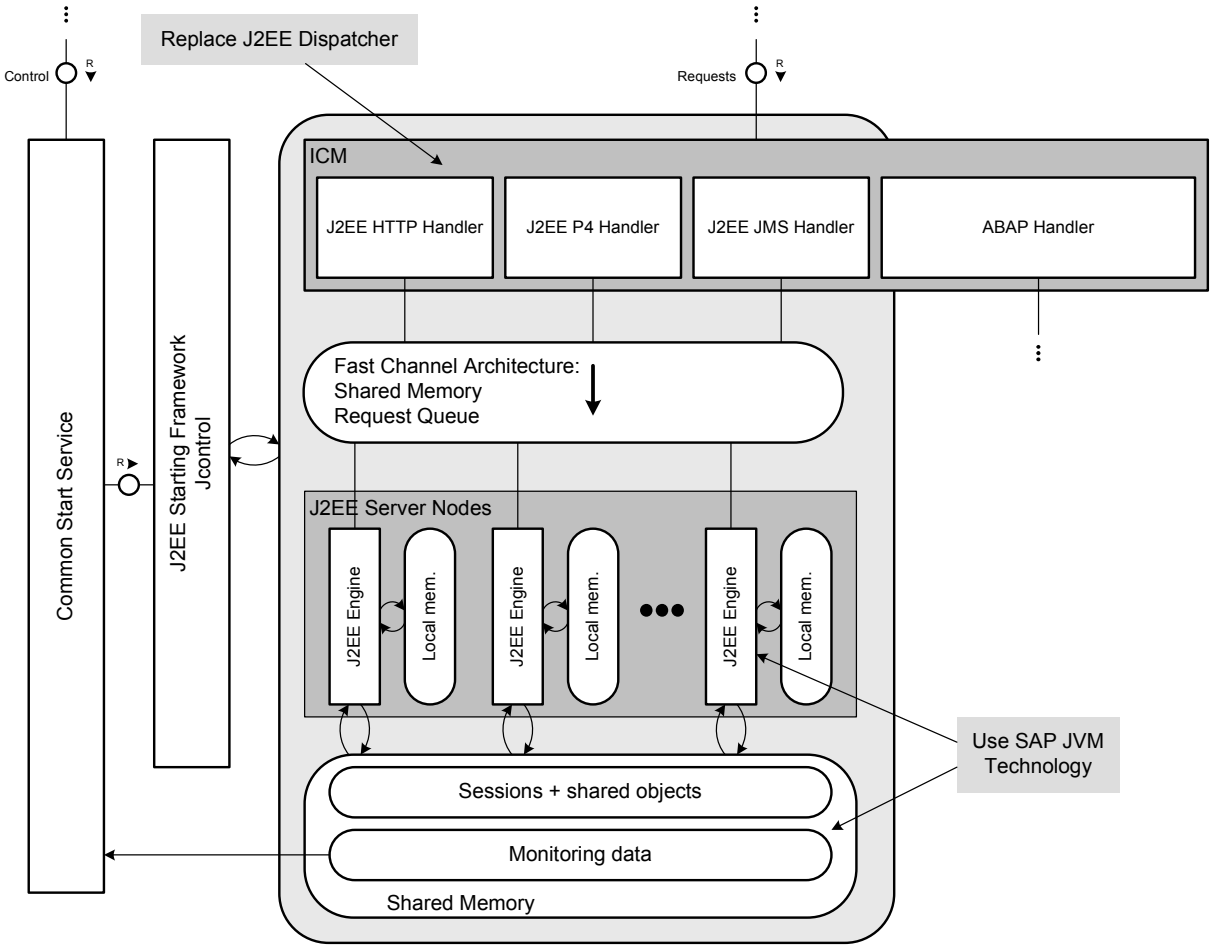

Figure 1: Architecture Overview Diagram (Static Structure at Runtime)

The Internet Communication Manager (ICM) process receives HTTP requests (also p4) and puts them into a Fast Channel Architecture (FCA) queue which is implemented using shared memory.

When a J2EE Node has sufficient resources to consume a request it will take the request out of the FCA queue, process it and write it back into the queue so it is returned to the originator of the request. All virtual machines have access to a shared memory region which is used as a session store and a commonly used cache. After every request, the server node will store the session associated with the request in the shared memory region in order to safeguard it against VM failures. Should the VM crash, another node can process the next request by reading the session information from the shared memory region.

Another part of the shared memory region can be used as a cache. The advantage is that commonly used data objects which are required by more than one session are stored in a single place (instead of storing it for every session) reducing memory consumption and access time.

At least a part of the architecture could have been implemented by standard java serialization. This approach is however 5 to 10 times slower than the shared memory approach implemented by SAP. This and the experience with shared memory are reasons for the approach taken.

All components in the upper layers (e.g. WebDynpro, SAP NetWeaver Portal, or NW applications) must adhere to certain rules and restrictions in order to be compliant to the architecture and benefit from it (in case they are not they will work as before but cannot benefit from this architecture). There is currently a Solid Rock adoption project which aims at making WebDynpro and the SAP NetWeaver Portal compliant to the Solid Rock architecture.

Total Cost of Ownership

The SolidRock project is part of the J2EE Engine. There are no separate TCO issues to be dealt with.

The architecture is designed to improve overall performance compared to the pre SolidRock architecture of the J2EE Engine. The performance improvement is achieved by using shared memory instead of socket communication.

Deployment

n/a

Maintenance

This architecture is part of the J2EE Engine.

Architecture Documentation

There are architecture documents and there have been several rollout presentations to the NetWeaver organization. Some public documentation can be found in the “Solid Rock Rollout” collaboration room (link https://portal.wdf.sap.corp/irj/index.html?NavigationTarget=CollaborationConnector://portal_content/com.sap.ip.collaboration/Rooms/f04f8dd9-b822-2810-92b7-d0446c6c418a/workset&NavTargetAsRoot=true).