

JBuddy™ SDK Scalability Report

Abstract

Enterprise applications have various scalability requirements. This report exposes some of the scalability issues developers face when developing scalable systems with the JBuddy SDK.

JBuddy SDK Background

The JBuddy Developer Studio (aka JBuddy SDK) is a versatile Instant Messaging Software Development Kit (Framework) written in Java™. Originally available only for Java developers, JBuddy for COM/.NET brings this robust framework to Windows COM and .NET developers. JBuddy SDK is used for IM-enabling existing applications as well as creating new client-side and server-side applications and systems that leverage both public and enterprise IM networks.

JBuddy Scalability Defined

Application Scalability refers to an application's capacity to handle a greater amount of something without serious negative consequences. This is measured differently for different applications. For example, IM server-side applications fall into several categories: JBuddy Alerts, JBuddy Bots, and JBuddy Gateways. For JBuddy Alert applications, a typical measure of scalability would be the system's capacity to send IM alerts. For JBuddy Bot applications, a typical measure of scalability would be the system's capacity to process IMs (both send and receive). For JBuddy Gateway applications, a typical measure of scalability would be the system's capacity to support concurrent users (total messages processes and probably more importantly, how many users online at a given time).

Scalability Variables

Several variables contribute to the overall scalability of a system leveraging JBuddy:

- Speed and number of CPUs
- Size and speed of system memory
- Operating System and version
- Java Runtime Environment and version
- Speed of internet connection
- IM service
- Efficiency of software exercising JBuddy

The overall system is only as good as it's weakest component. Therefore, in order to maximize scalability of the system, each of the variables should be carefully configured in conjunction with the other variables to provide a balanced, efficient system. Given the number of the variables, this report will focus on the critical areas pertaining to IM and JBuddy.

JBuddy™ SDK Scalability Report

CPUs and Memory

For JBuddy Alert and JBuddy Bot applications where one or only a small number of clients are responsible for sending a large number of messages quickly to many other clients, the efficiency of the software exercising the JBuddy SDK and the speed of the CPUs and the availability of adequate memory to avoid swapping to disk or paging are the most critical variables in scalability.

Operating Systems, Java Runtime Environments

As of this writing, the current framework (JBuddy SDK version 2.4, January 2005), runs within a Java Runtime Environment and supports Sun's JRE 1.3.1+, 1.4.1+, and 1.4.2+. For JBuddy Gateway applications where the system should support a large number of concurrent users, the Operating System and the JRE are critical variables in the overall scalability of the system. Operating systems typically have a limit on their capacity to manage concurrently executing tasks (processes or threads) and a limit on their capacity to managed open files or network connections (sockets). Operating systems usually need special configuration in order to reach their maximum capacity in these critical areas. Some operating systems are only limited by the amount of available physical memory, but at some point their capacity is still limited by their CPU's capacity to handle all the processes and sockets with enough efficiency to keep the overall system responsive enough.

Client/Server and JBuddy IM

In a typical client/server architecture, you have one or more clients and one or more servers. In a JBuddy-enabled solution, the JBuddy SDK provides the framework to create IM clients (IClient objects) and connect (IClient.connect) them to an IM service, be it internal (Enterprise IM), or external (Public IM Service). For JBuddy Gateway applications, there will be hundreds, if not thousands of clients that need to be connected to the IM service.

Currently, the JBuddy SDK has some limitations based on the use of system resources. Each client (screenname) created by the framework uses one listener thread and one socket. In the case of the MSN and Lotus Sametime protocols, one socket and one thread is spawned for each 'conversation' between the client and the 'buddy'. In the case of the XMPP (Jabber) and JSC (JBuddy Message Server) protocols, clients created by the framework use two threads for each client, one event thread and one listener thread. Additionally, clients can send and receive files from other users if enabled and the file transfer causes another thread to be spawned and another socket to be opened, one for each file transfer session.

JBuddy Scalability Conclusions

No attempt will be made to pinpoint how many messages a JBuddy Alert or JBuddy Bot application can process per second. No attempt to pinpoint how many concurrent users a

JBuddy™ SDK Scalability Report

server can support for a given IM service within a JBuddy Gateway application. Instead, this report will focus on drawing conclusions based on similarities between a JBuddy-enabled application and prior work documented in the Volano Report, Copyright © 2003 Volano LLC. Please refer to this report online at <http://www.volano.com/report/>

As stated above and well documented in the Volano Report, the number of messages that can be processed per second and the number of concurrent users supported varies widely depending on the Operating System, the JRE version and the configurations of each. Based on the volano report, and all other factors the same, the JBuddy SDK should more than adequately handle high volume alert and bot applications (assuming IM rate limits lifted by service agreements with IM service providers). The JBuddy SDK should support approximately 4000 concurrent IM users and conversations on the Sun JRE 1.4.1 on a Solaris 8 machine and should support approximately 3000 concurrent users and conversations on a Windows 2000 SP3 Server, as configured in the Volano Report.

Notices

Copyright © 2005 Zion Software, LLC. JBuddy and JBuddy SDK are trademarks of Zion Software, LLC. The Volano Report is copyright Volano LLC, Java is a registered trademark of Sun Microsystems, Inc. All other names and trademarks are the property of their respective owners.