High availability and scalable platform for Java/C++ on Solaris

Asked 16 years, 3 months ago Modified 8 years, 5 months ago

Viewed 2k times

9

I have an application that's a mix of Java and C++ on Solaris. The Java aspects of the code run the web UI and establish state on the devices that we're talking to, and the C++ code does the real-time crunching of data coming back from the devices. Shared memory is used to pass device state and context information from the Java code through to the C++ code. The Java code uses a PostgreSQL database to persist its state.

We're running into some pretty severe performance bottlenecks, and right now the only way we can scale is to increase memory and CPU counts. We're stuck on the one physical box due to the shared memory design.

The really big hit here is being taken by the C++ code. The web interface is fairly lightly used to configure the devices; where we're really struggling is to handle the data volumes that the devices deliver once configured.

Every piece of data we get back from the device has an identifier in it which points back to the device context, and we need to look that up. Right now there's a series of shared memory objects that are maintained by the

Java/UI code and referred to by the C++ code, and that's the bottleneck. Because of that architecture we cannot move the C++ data handling off to another machine. We need to be able to scale out so that various subsets of devices can be handled by different machines, but then we lose the ability to do that context lookup, and that's the problem I'm trying to resolve: how to offload the real-time data processing to other boxes while still being able to refer to the device context.

I should note we have no control over the protocol used by the devices themselves, and there is no possible chance that situation will change.

We know we need to move away from this to be able to scale out by adding more machines to the cluster, and I'm in the early stages of working out exactly how we'll do this.

Right now I'm looking at Terracotta as a way of scaling out the Java code, but I haven't got as far as working out how to scale out the C++ to match.

As well as scaling for performance we need to consider high availability as well. The application needs to be available pretty much the whole time -- not absolutely 100%, which isn't cost effective, but we need to do a reasonable job of surviving a machine outage.

If you had to undertake the task I've been given, what would you do?

EDIT: Based on the data provided by @john channing, i'm looking at both GigaSpaces and Gemstone. Oracle Coherence and IBM ObjectGrid appear to be java-only.

java c++ scalability solaris high-availability

Share

Improve this question

Follow

edited Jul 17, 2016 at 8:49

Martijn Pieters

1.1m ● 319 ● 4.2k ● 3.4k

asked Sep 9, 2008 at 5:52

Andrew

12k ● 12 ● 71 ● 85

3 Answers

Sorted by: Highest score (default)

5

The first thing I would do is construct a model of the system to map the data flow and try to understand precisely where the bottleneck lies. If you can model your system as a pipeline, then you should be able to use the theory of constraints (most of the literature is about optimising business processes but it applies equally to software) to continuously improve performance and eliminate the bottleneck.

Next I would collect some hard empirical data that accurately characterises the performance of your system. It is something of a cliché that you cannot manage what you cannot measure, but I have seen many people

attempt to optimise a software system based on hunches and fail miserably.

Then I would use the Pareto Principle (80/20 rule) to choose the small number of things that will produce the biggest gains and focus only on those.

To scale a Java application horizontally, I have used Oracle Coherence extensively. Although some dismiss it as a very expensive distributed hashtable, the functionality is much richer than that and you can, for example, directly access data in the cache from C++ code .

Other alternatives for horizontally scaling your Java code would be Giga Spaces, IBM Object Grid or Gemstone Gemfire.

If your C++ code is stateless and is used purely for number crunching, you could look at distributing the process using ICE Grid which has bindings for all of the languages you are using.

Share Improve this answer

Follow

answered Sep 9, 2008 at 9:00

John Channing

6,591 ● 7 ● 47 ● 57

1

You need to scale sideways and out. Maybe something like a message queue could be the backend between the frontend and the crunching.

Share Improve this answer

Follow

answered Sep 9, 2008 at 6:05

Eugene Yokota

95.5k ● 45 ● 217 ● 320

1

Andrew, (in addition to modeling as a pipeline etc), measuring things is important. Have you ran a profiler over the code and got metrics of where most of the time is spent?

For the database code, how often does it change ? Are you looking at caching at the moment ? I assume you have looked at indexes etc over the data to speed up the Db ?

What levels of traffic do you have on the front end ? Are you caching web pages ? (It isn't too hard to say use a JMS type api to communicate between components. You can then put Web Page component on one machine (or more), and then put the integration code (c++) on another, and for many JMS products there are usually native C++ api's ie. ActiveMQ comes to mind), but it really helps to know how much of the time is in Web (JSP ?) , C++, Database ops.

Is the database storing business data, or is it being also used to pass data between Java and C++ ? You say you are using shared mem not JNI ? What level of multi-threading currently exists in the APP? Would you

describe the code as being synchronous in nature or async?

Is there a physical relationship between the Solaris code and the devices that must be maintained (ie. do all the devices register with the c++ code, or can that be specified). ie. if you were to put a web load balancer on the frontend, and just put 2 machines up today is the relationhip of which devices are managed by a box initialized up front or in advance?

What are the HA requirements ? ie. just state info ? Can the HA be done just in the web tier by clustering Session data ?

Is the DB running on another machine ?

How big is the DB ? Have you optimized your queries ie. tried using explicit inner/outer joins sometimes helps versus nested sub queries (sometmes). (again look at the sql stats).

answered Sep 9, 2008 at 12:29

nso1

605 ● 10 ● 18