Dynamic Intelligent Application Delivery in a Distributed Environment

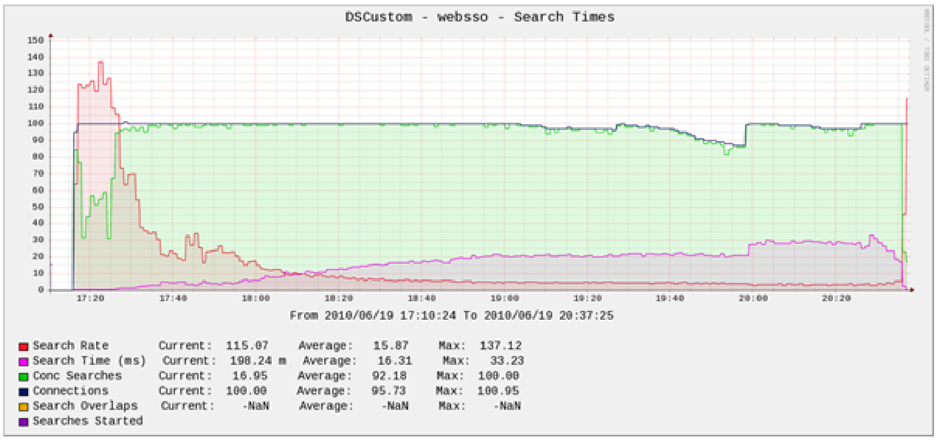
Hamish, 2010-21-09

Designing for fine grained control and monitoring in a load balanced environment – Part I – An initial simple approach

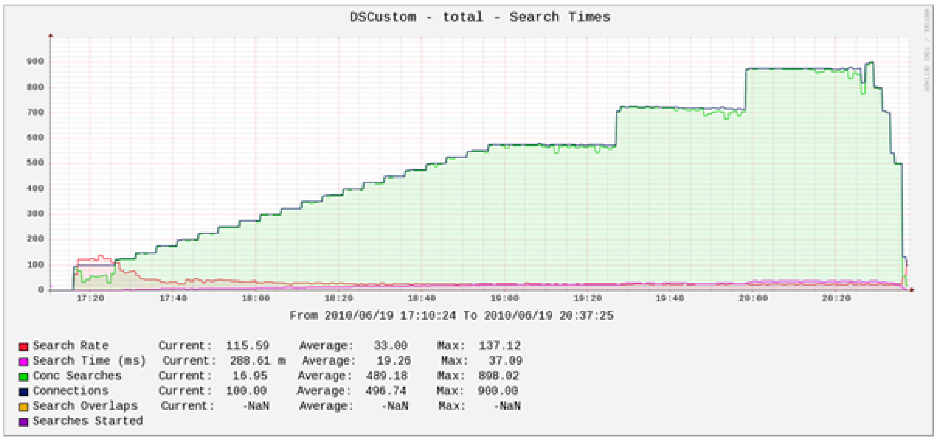
The Reason to Control

In many environments, scaling the service has typically been the answer toward keeping response times down as load grows ever larger. However there becomes a point when simply scaling the service sideways reaches the point where the overheads become significant. Especially in a financial industry situation where you may have 1 or even 2 dark sites on idle standby for DR.

However, even keeping up with the scaling can affect us in times of exceptional load. When loads increase beyond the capacity of the service to accommodate them, we get the scenario shown in the graph below. A steady number of connections and searches from a high priority application shows a steady increase in search response times, and a corresponding increase in the number of concurrent searches over time.



The why is shown below. As more and more users attach to the service, we can see that performance suffers for those users that are regarded as critical to the organisation.



How to Solve it

One of the ways to solve these challenges, is with dynamic intelligent load balancing.

By adding intelligence into the dynamic load balancing available from a BigIP ADC, we can measure the demand and the response times and adjust the load distribution dynamically in near-realtime, using intelligence to predict the outcome of the changes being made.

This intelligence is implemented using a number of measuring, reporting and controlling processes, built around a central database and statistics store (mySQL and RRD).

Why Use Intelligent Load Prediction

The answer is to avoid state flapping. A problem inherent with feedback systems is the tendency to make system changes that directly affect the statistics being used to provide the decision. As the changes are made, the statistics move below the threshold and the system is changed again. This hunting effect produces a system oscillation.

This is a consequence of automation towards simple changes with complex conditions and can be demonstrated with a single high priority and many low priority users when a simple measurement is used to indicate when the system should be in NORMAL or CRITICAL mode and no prediction is made for the results of the changes introduced by the controller.

If we simply used the response time of a high priority application to decide whether to run in NORMAL or CRITICAL modes then the very outcome of moving to CRITICAL where high priority traffic is expected to regain suitable response times would then trigger a shift back to NORMAL mode (Which would then result in response times increasing again).

The Results of Performing Simple Dynamic Load Shifting

We can compare the ‘websso’ response time graphs for a similar increasing load when dynamic alterations of the load balancing are being made. Although the concurrent search count increases, the concurrent searches and search times remain constant for the high priority traffic (WebSSO) as the total load increases.

The Methodology

The method of load shifting using a split pool approach was chosen for this test in conjunction with monitoring the backend server loads.

Monitoring Load

The load of each member in the server pool was monitored and combined to provide a total load for the whole pool. Statistics to calculate server load are gathered by a dedicated process gathering published counters for SunONE Directory server performance via LDAP queries.

The load for individual servers were then gathered by the managing service so that the load across the entire pool of servers could be computed.

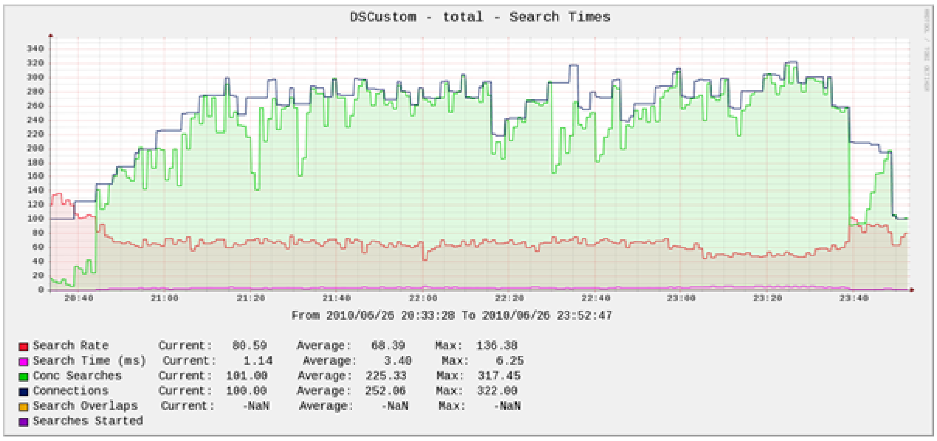
Split Pool Processing

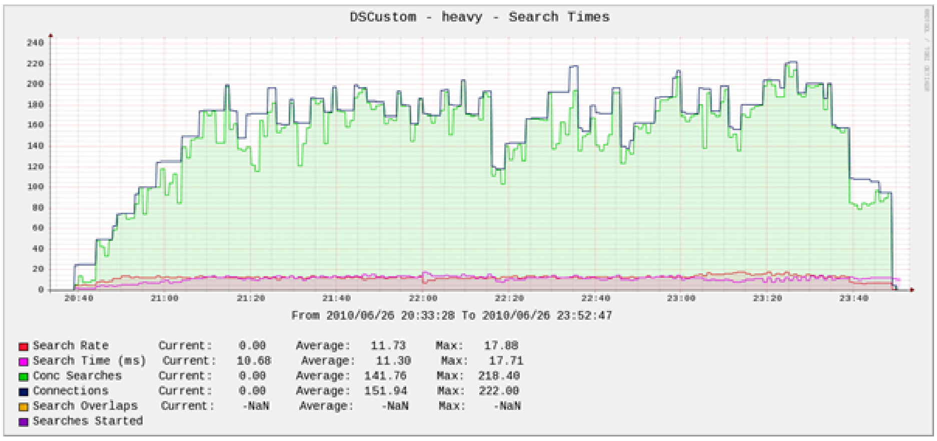
Given a number of servers in NORMAL mode being used to service all the traffic on the service, we monitor the server loads for each backend and gather those stats to computer the pool loads across all servers. When the load of the default pool exceeds a defined level, we split the default pool of servers into a high priority and low priority pool in a 2:1 server ratio. The high priority traffic is then routed to the high priority servers. Detection of high priority traffic was performed by decoding the userid from LDAP bind requests in a BigIP iRule, and utilising a table of users held in a BigIP DataGroup.

The pools in this example are ldappool1 (NORMAL mode default pool), ldappool_hipri (CRITICAL mode high priority traffic) and ldappool_sinbin (CRITICAL mode default pool).

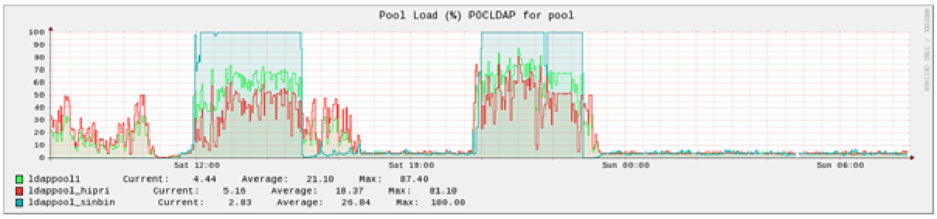
The Results

We can see as a result of the traffic splitting that the initial search rates start high and drop similar to the previous example as total load increases. However we can see that search rates then bottom out and remain relatively high, even as the total load on the system continues to increase.

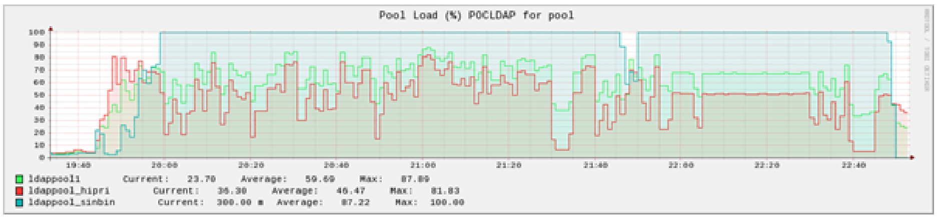




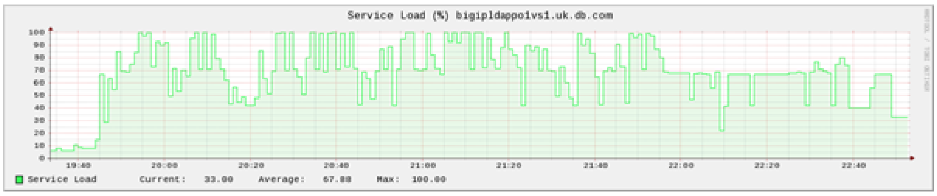
The pool load graph for the day in question helps to explain what's happening here...



And in a more detailed version covering the same time period as the previous graphs we can see how the load is being shifted from the default pool (ldappool1) to the ldappool_sinbin pool. And how this keeps the load down on the ldappool_hipri pool



The dynamic routing of users to specific pools is being triggered by the calculated service load (Calculated from the load on individual nodes). The values used for this run being > 40% on ldapppool1 trigger to CRITICAL and < 20% trigger (On ldappool1) to NORMAL. Between 20-40% is a hysteresis region where the mode remains as it was (So for example on an increasing load 30% may be NORMAL while on a decreasing load 30% may be CRITICAL if the boundaries have been triggered previously.



Note that these graphs also show a limit on the number of simultaneous connections that the heavy userload was able to attain. This is due to resource limits on the slapd servers being used at the time.

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com