**Testing Hydstra Layout – What Database, Timeseries and Temppath Locations will give optimal performance**

**IDEAS PAPER. INITIAL STARTING POINT . Friday, 17 January 2014**

Background:

Hydstra evolved initially as a single user set of applications running on a single. Most installations use FoxPro tables for metadata such as sites and rating tables, Time series data is kept in binary blob files outside the database tables. As hydstra applications run they produce temporary files and output files. These typically are written to the users local Temp directory.

In corporate environments Hydstra is normally shared by a small workgroup on a local network server. As the organisation scales in numbers or in geographic spread various decisions are made with respect to how these core components are located and/or distributed.

At NOW the solution involves locating the Hydstra system in the corporate data centre with access via a Citrix farm. The Hydstra components are accessed via a network mapping to a single server. This architecture was chosen according to a division of responsibility where the IT service provider manages the Citrix farm and the Hydrometric systems team manage the Hydstra server.

Various NOW users, for various purposes, also maintain standalone systems based on the single machine architecture. Additionally various auxiliary Hydstra systems/servers exist for special purposes or troublesome applications.

The simplest single machine architecture and the NOW Architecture are shown in the following 2 diagrams.

|  |
| --- |
| Single Machine: |
|  |
| SF-Citrix and NOW-Hydstra: |
|  |

Issues:

NOW is upgrading it’s Hydstra system. We have 2 issues to contend with.

1. In simple comparisons the performance of the Citrix system is significantly worse than the Standalone systems. Also when users accesses the Hydstra system directly via remote desktop there are levels of performance gains over accessing the same system via the network mapping in Citrix.
2. NOW needs to move the Hydstra system to a set of new server and move the databases to MS SQL Server and we don’t want that change to make the performance any worse. In fact we want to provide a better user experience.

Implementing on SQL Server

|  |
| --- |
| The situation gets more complicated when adding SQL Server as ‘corporate policy’ suggests that it should use the corporate SQL Server which is located further away on network. |
|  |

We are intending to test both the Corporate SQL and the implications of running a local SQL version.

Approach

1. A matrix of (16) combinations of the basic components has been created.
2. These have be implemented as Hydstra systems and initialised via HyMulti.
3. A script has used the IniFile for Hymulti to automate testing using HyTest .
4. The results of the different tests are being analysed to determine what architectural attributes influence the performance. This analysis will guide the implementation.

The Configuration Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| code | Database | TimeSeries | Temppath |
| lfp-lts-lt. | local FoxPro | local TimeSeries | local TempPath |
| lfp-lts-mt. |  | local TimeSeries | mapped Temppath |
| lfp-mts-lt. | local FoxPro | mapped TimeSeries | local TempPath |
| lfp-mts-mt. |  | mapped TimeSeries | mapped Temppath |
| mfp-lts-lt. | mapped Foxpro | local TimeSeries | local TempPath |
| mfp-lts-mt. |  | local TimeSeries | mapped Temppath |
| mfp-mts-lt. | mapped Foxpro | mapped TimeSeries | local TempPath |
| mfp-mts-mt. |  | mapped TimeSeries | mapped Temppath |
| lsql-lts-lt. | local SQL Server | local TimeSeries | local TempPath |
| lsql-lts-mt. |  | local TimeSeries | mapped Temppath |
| lsql-mts-lt. | local SQL Server | mapped TimeSeries | local TempPath |
| lsql-mts-mt. |  | mapped TimeSeries | mapped Temppath |
| nsql-lts-lt. | network SQL Server | local TimeSeries | local TempPath |
| nsql-lts-mt. |  | local TimeSeries | mapped Temppath |
| nsql-mts-lt. | network SQL Server | mapped TimeSeries | local TempPath |
| nsql-mts-mt. |  | mapped TimeSeries | mapped Temppath |

Machine Configurations:

Initial testing of these configuration has been on 2 workstations

Machine 1: Win 7: 12 Gb Ram Xeon 3.GHz

Machine 2: Win 2012 R2 server vm 8GbRam 4 Virtual Processors

LAN private 1GByte – unrealistically fast for datacentre – for some reason.

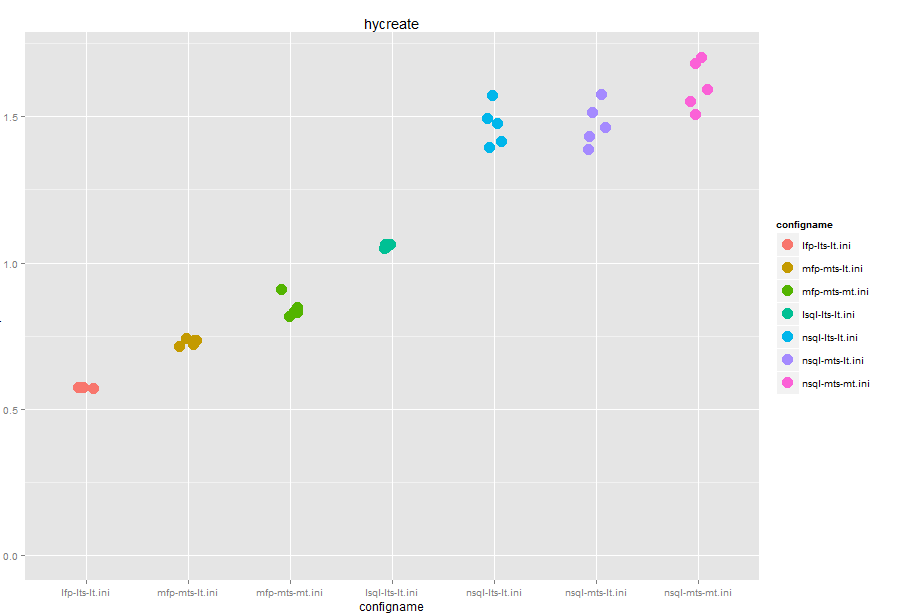
Move to testing in Data Centre ASAP. To get closer to real environment.

Plan

* The lfp-lts-lt will be the simplest/fastest – a given.
* The mfp-mts-mt is the configuration closest to what we have on Citrix.
* We want to move to SQL. It may be either local to Hydstra server(lsql-mts-mt) or provided on a networked SQLServer (nsql-mts-mt) (preferred for administration).
* Test what applications run better in what configuration and understand how to get the best performance while delivering the applications in a corporately supported arrangement.
* Determine how to get better LAN in the Data centre
* Determine how to feedback to Kisters so that get Applications tuned for SQL Server rather than a generic application.

Examples:

1, single simple Hycreate Job: 5 repeats of each job. time in seconds. (0 - 1.75)



Notes:

Takes 06 second to do this hycreate on lfp-lts-lt. Each addition of mapped TS and mapped Temp add time.

The current mapped FP and mapped TS and Mapped Temp takes about .9 sec to do a Hycreate. 30% slower.

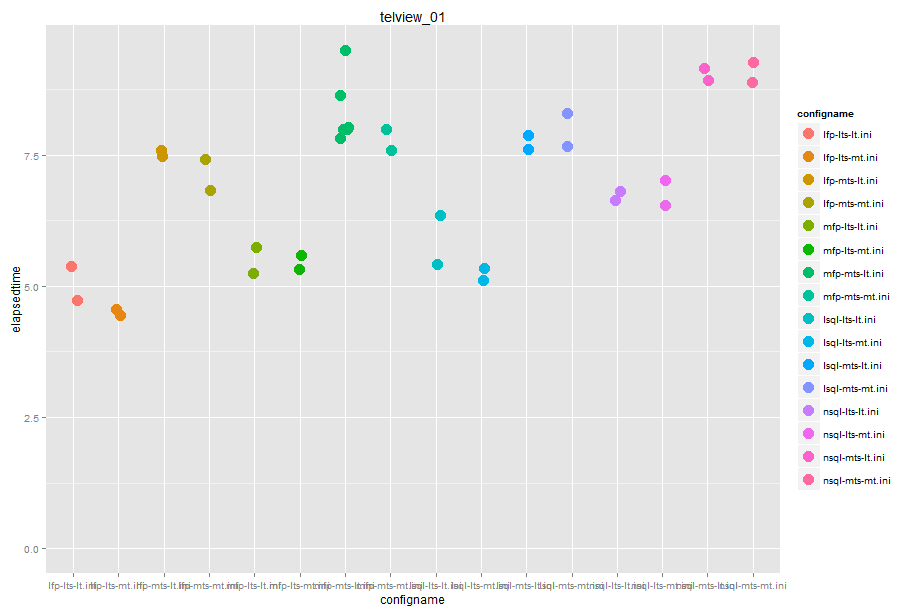
Moving to a local system based on SQL Server is about twice as slow as local FoxPro system.

The move to Mapped TS files, mapped Temp and a network SQL system is about 1.5 sec or 2.5 times as slow as a lfp-lts-lt system.

1. Telview: Loading Telview with a Hydro’s site list of Prov data, then closing it down.

Goal to measure the start up time.

Time scale 0 – 10 sec



Notes:

Local lfp-lts-lt system system takes 5 seconds.

If move to mapped Foxpro there was no real impact to load time.

When moved to Mapped Timeseries there is a 2.5 second or 50% increase in the load time.

When all the components are on a mapped drive running at about 8 sec. or a 60% increase in load time.

Going to SQL in this case gave a further but marginal increase in load time, even when the SQL was on a networked drive.