Big Data with SAP HANA

(Original creator: JamesRodger)

I’ve been interested in large scale computing ever since I was introduced to it at the University of Southampton where the Computer Science department was heavily involved in Data Mining and Grid Computing research, which obviously influenced the courses on offer and what the lecturers liked to talk about. My dissertation looked at how these techniques could be applied to protein folding research, which generated much more data than could be handled effectively with the technologies available at the time. We are producing more and more data at an ever increasing rate each day, so these challenges are becoming relevant to more and more businesses and sectors. Of course things have moved on in recent years. The buzz words today are Cloud Computing and Big Data, evolutions on those older concepts that I’ve followed with great interest.

Given this I was happy to recently have the chance to do some research into SAP’s in-memory database solution HANA (High-performance Analytic Appliance, a somewhat tortured acronym). Clearly it’s a complex platform and I’ve only scratched the surface of what it’s capable of. It’s tricky to sum up, but to quote from a SAP HANA white paper (http://www.saphana.com/blogs/experts/2012/02/01/latest-sap-hana-whitepaper-reveals-more-than-just-technical-details):

"The SAP HANA® database is an in-memory database that combines transactional data processing, analytical data processing, and application logic processing functionality in memory. SAP HANA removes the limits of traditional database architecture that have severely constrained how business applications can be developed to support real-time business."

This blog post is a good place to start if you would like to know more about SAP HANA: http://www.saphana.com/community/blogs/blog/2012/09/07/sap-hana-for-beginners

One of the benefits that really jumped out at me was the ability to query for complex aggregated data in real time. With a standard database it can take too much time to fetch, for example, the balance of a financial ledger. Because of this it’s left to the application layer to run batch jobs which calculate these figures and store them in additional tables. HANA allows you to delegate this task back to the database layer. We can therefore simplify our data models, we don’t have to write specific application code to handle the task and our totals are truly “real time” because we do not have to run nightly batch jobs to update them. This leaves our application to focus on what it should be doing, implementing business logic and not worrying about how to overcome the limitations of the database platform.

Also of note is that HANA provides a standard SQL interface over ODBC, this means that existing applications can start to use its functionality with a minimum of migration effort. In businesses utilising HANA, Uniface would be able to play a crucial role in tying together various different applications and technologies they might have. One use case might be for a Uniface interface able to pump real time data into HANA from other disparate applications. Since a HANA database is likely to be structure differently to a traditional database Uniface’s powerful ability to transform and map data would be invaluable to the process.

It’s fairly simple to try out HANA. SAP provides a very handy 30 day developer trial on CloudShare (http://scn.sap.com/docs/DOC-28191). Once you’re signed up it’s just a case of logging into CloudShare and then starting up the VM environment. You’re then free to do whatever you like with the servers. In my case I uploaded a copy of Uniface, ran the ODBC configuration utility to setup my driver settings and had a play around. Incidentally, I can highly recommend taking an initial snapshot of your VMs so that you can revert to it in case you completely break the servers (like I did).