

Re: Borland is neglecting Delphi for Win32 badly

Source: <http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.non-technical/2005-01/6244.html>

From: Euan Garden (euangarden@hotmail.com)

Date: 01/20/05

Date: Wed, 19 Jan 2005 21:42:14 -0800

> *Are you saying, then, Euan, that pure performance isn't the only thing
> you've considered when building Yukon? Things like safety, security,
> and a controlled execution environment are also valuable features?*

Absolutely. I am sure if you were to poll the SQL team SQLCLR means different things to different people, for me there are 2 keys things it means.

1/ Choice

This is about language choice and also architectural choice. Today in SQL Server 2000 you can write server extensions as COM objects and call them via sp_Oa, but we do not recommend it in high workload/availability environments. We also allow you to use XPs, 99.9% of these are written in C++. This means we are preventing entire classes of dev shops from extending their architectures using server side logic other than T-SQL. SQLCLR allows CLR based languages to be used depending on the nature of the IL that they generate.

To give an idea here of how simple this can be, when we worked on the demo for Borcon 2 years ago of code executing in the server, I was exchanging mail with Borland folks 4/5 times a week for weeks, they were constantly updating dccil, we were integrating new builds of Whidbey, debugging was a HUGE pain, but we got it working on one machine 4/5 days before, with a Franken build from Borland and SQL. This year, the demo for the MS keynote consisted of me installing the shipping Delphi 8 on a daily build of SQL 2005, doing a quick test and then getting on a plane. Danny and I spent 5 mins tweaking in the lab when I arrived, and there it was, a 50x perf bump. Lets hope its that easy for any language at RTM.

Ok so we solved the language choice problem, what about the architecture barrier? We have tried to keep the programming models for inproc and out of proc as close as possible, there are differences but we believe they are acceptable. You should be able to take a datatype that runs on the middle tier today and extend the server with it in SQL2005, this could reduce serialisation and mapping complexity, oh and the same is true for Procedures and Functions, triggers and aggregates are a little more database specific.

2/ Modern execution environment.

We get so much from hosting the CLR in this space, its worthy of a mammoth blog entry but I'll keep it short. There are some real challenges with the XP architecture today, by default XPs use the MemToLeave memory segment from SQL Server, this is not managed by our normal memory manager and when subjected to leaks and fragmentation can cause serious side effects. There is also a problem with developers being able to write whatever code they want in an XP, they can call UI (hanging the server as if its started as a service rather than from the console it has no handle to the desktop) they can use synch primitives (causing side effects on UMS, the User Mode Scheduler). By moving to IL we can verify what is coming, and optimise based on that knowledge, for example we bounce assemblies that call UI (Win32 or Web) in Safe and External Access mode, likewise if you allocate memory directly or use sync primitives.

The security benefits are also tremendous and we could discuss for a long time the benefit of a scalable multi platform environment, we have a weekly meeting on stress and perf between SQL and CLR teams, 2 years ago we talked about 2 way 32 bit perf, now we are discussing perf on 64 ways using IA-64...

Can you tell I love this particular feature? While perf makes a great demo the real payback from the CLR integration with SQL Server will come in the days and years that follow that initial deployment in the security, scalability, ease of maintenance and reliability of production systems, the same can be true for any .Net app.

-Euan

"Nick Hodges [TeamB]" <nickhodges@gmail.com> wrote in message news:41ef21e3\$1@newsgroups.borland.com...

> *Euan Garden wrote:*

>

>> *We have integrated the CLR very deeply into SQL2005 because it gives us a safe, secure, controllable execution environment, by doing that it is closer to the data and system components.*

>

> *Are you saying, then, Euan, that pure performance isn't the only thing you've considered when building Yukon? Things like safety, security, and a controlled execution environment are also valuable features?*

>

>

>> *Oh and those aspects*

>> *of the CLR seem to be aspects you have forgotten in the discussion.*

>

> *Right -- there are things to consider /other/ than performance.*

>

> --

> *Nick Hodges -- TeamB*

> *Lemanix Corporation -- <http://www.lemanix.com>*

> *Read my Blog -- <http://www.lemanix.com/nick>*