Ingres is the original Open Source RDBMS

# Very complete and highly predictable

Roy Hann

**After Red Hat, which is the world's largest independent open source company? Here's a hint: it's the one that generates the largest revenue stream from an SQL DBMS – and grew its revenue by 32% in 2008.**

The answer, which might not be what you expect, is Ingres Corporation (www.ingres.com), the company created in November 2005 by venture buyout specialists Garnett & Hellfrich when they bought Computer Associates' Ingres business unit. Since being established with a several dozen key employees in 2005, Ingres Corporation has now grown to over 300 employees today, just over half of whom are actively involved in technical development.

## Eventful

Ingres has been around for years; it was the original open source RDBMS, released in 1975 long before the term "open source" was ever coined. It was massively influential and has had numerous spin-offs, some of which are very familiar names today, and of course Ingres itself is now open source again. The full story of Ingres is long and eventful and anyone who is curious can spend a few minutes reading the reasonably accurate account on Wikipedia. But why make a fuss about Ingres in 2009?

The reason for the surge of interest in the last couple of years is, of course, that businesses and governments all over the world are trying to stretch their budgets to do more with less. The allure of low-cost pay-as-you-go open source software is compelling. The fact that Ingres delivers proven high-end technical features and worldwide follow-the-sun support, as well as military-strength security and compliance features, makes it the obvious first choice SQL DBMS for government and commerce, particularly in the European Union. For instance in Ireland the Irish Revenue Commissioners (the national taxation body) is the largest user, with over 5,000 users just on their LAN, plus a publicly accessible web presence. Barclays bank in the UK (my bank incidentally, and still one of the largest in the world) also runs its core systems on Ingres. So does Lufthansa. And now so does Les Salins du Midi, and about 10,000 other paying enterprise customers.

Of course any IT worker over the age of about 40 will remember when Ingres dominated the market for Unix-based relational databases in the 1980s, but he will also remember the disappointing management that saw its market share eroded and the product lose prominence. From the point of view of the successive proprietors trying to make money, higher visibility would have been nice, but of course Ingres was already widely deployed and was quietly getting on with running the core systems of banks, utilities and all levels of government.

## Highly multithreaded

As a result, for more than a decade Ingres steadily enhanced the essential, if not attention-grabbing features that are the difference between a simple SQL engine and a robust bet-the-business DBMS. This attention to critical basics back then means that today Ingres finds itself in the happy position of being described in a research note produced by Gartner Inc. last November, as the only open source DBMS with the functionality to support mission critical systems. And in another research note the following month Gartner puts Ingres in its magic quadrant of data warehouse DBMSs.

At this point, for those of you to whom Ingres is not a familiar name, it is worth taking a brief look at the architecture of Ingres. Ingres is actually a collection of cooperating, multithreaded server processes. On platforms that support POSIX threading – and these days that includes even OpenVMS – the central Ingres server process (*iidbms* in figure 1) is highly multithreaded. There are approximately 15 threads responsible for various supervisory and house-keeping tasks; the exact number can be adjusted as required by the Ingres system administrator. In addition, each user session runs as a separate thread in the server. And even within each session, further threads will be spawned during the execution of a query so that, for example, the various partitions of a partitioned table can be processed simultaneously, in parallel. In addition, independent parts of a complex query can also be processed in parallel in separate threads. And in fact, successive steps of even a simple query can run in parallel: as soon as rows are available from a preceding step they are supplied to the next step without waiting for the earlier step to complete its processing. All these threads get distributed over all the available CPUs and cores automatically, so that without any programming by the application programmer, the entire resources of the host hardware get fully exploited.

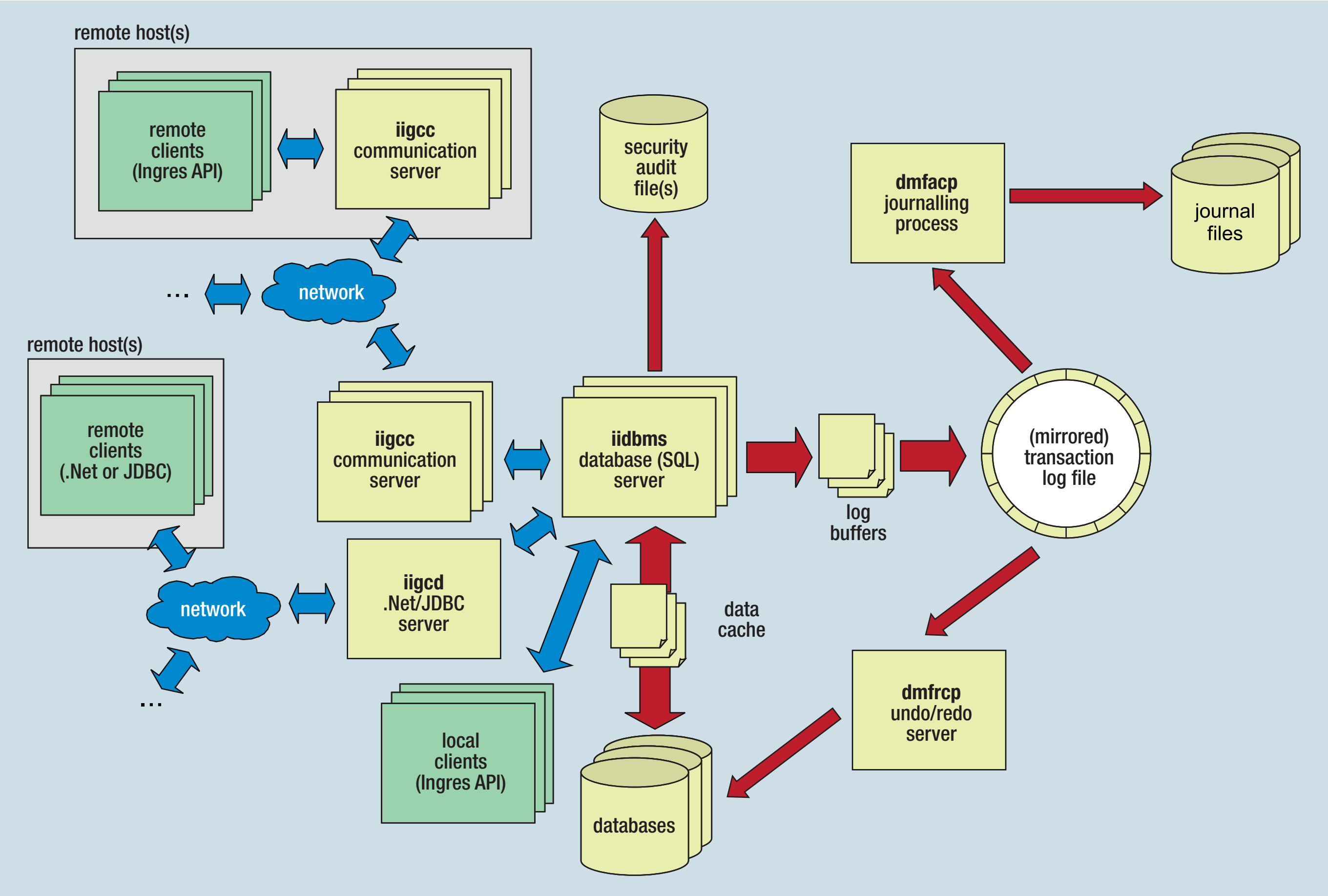
## Tuning

I mentioned above that the Ingres system administrator can adjust the number of threads used for supervisory tasks. Ingres has long had a reputation for being pretty much self-managing once it is installed and tuned for the local requirements, but when first installed it may need to be tuned for the local hardware resources. A useful feature of Ingres system administration is that the system administrator doesn't need expert knowledge to be able to make effective tuning changes. There is a handful of key parameters, including for example one called *connect_limit*, which a relatively naïve system administrator can change. By changing one of these key parameters, a cascade of related (or derived) changes is made automatically, based on built-in configuration rules that define how other parameters should be scaled in proportion. Over time, as the system administrator becomes more adept at tuning Ingres, it is of course possible to adjust other lower-level parameters for more precise control, and these adjustments too may set off a cascade of yet more related changes. Ultimately, a guru-level system administrator can adjust even the configuration rules.

It is also worth noting that there doesn't even have to be a system administrator at all to make these adjustments. If one is developing a shrink-wrapped system for unattended deployment in a remote area, the adjustments can be scripted using the *iisetres* utility, and exactly the same consequent derived parameter settings will be made as if working interactively using VCBF.

## Multiple installations

The diagram in figure 1 shows a single, simple Ingres installation. A single host machine can in fact host multiple Ingres installations. Typically one might have a production installation and a totally independent development installation on the same


remote host(s)
remote clients (Ingres API)
iigcc communication server
security audit file(s)
dmfacp journalling process
journal files
...
network
remote host(s)
remote clients (.Net or JDBC)
iigcc communication server
iidbms database (SQL) server
log buffers
(mirrored) transaction log file
network
iigcd .Net/JDBC server
data cache
dmfrcp undo/redo server
...
local clients (Ingres API)
databases

**Afbeelding 1:** Ingres System Architecture.

machine. (I sometimes like to configure my development installations with minimal resources so that the programmers quickly notice inefficient code.) Another common practice is to have a test installation on the same machine, running the next release of Ingres for evaluation purposes before deploying it in production. Still other sites have multiple installations each supporting a specific department.

The ability to have separately managed installations is highly convenient if one wants to stop and restart one without affecting all other users, or to be able to move an installation and its databases to a physically different host machine. Of course sometimes it is also necessary to have access to a database in one installation from another installation. In this case, whether the database is in another installation on the same machine, or another host somewhere else on the network, the Ingres communication architecture conceals the physical location from the application programs so that communication is always seamless. This can be taken a step further. The system administrator can use an Ingres Star server to create a single logical database, consisting of multiple real databases, possibly in multiple installations on multiple hosts, which can be updated reliably using the two-phase commit protocol completely transparently. What is more, the logical database doesn't need to consist solely of Ingres databases, because it can also include tables from other DBMS products such as DB2 and Oracle by the using the appropriate Ingres gateways.

A single Ingres installation can contain an unlimited number of databases. (An Ingres database corresponds to the ANSI/ISO concept of a catalogue, as does an Oracle database or a DB2 database.) An Ingres database can contain an unlimited number of schemas too, so each user can have their own schema. Each schema can contain an unlimited number of tables, views, and so on. Physically, Ingres tables can be stored in files or in a raw table space. When using files, a single table can be striped over many files. One of the best reasons to do this is so that the stripes can be backed up in parallel, which is a major time saver for large tables. It also allows a table to be bigger than the biggest file size supported by the operating system. The files can be compressed too. Tables can also be partitioned into as many as 65,000 partitions using any combination of four types of partitioning rule to partition and subpartition the table, and each partition can be striped over multiple files.

## Standards

One of the first things that the new user notices about Ingres is that by default everything you do is fully transactional, including all DDL. So not only are individual updates to rows in tables capable of being rolled back, so are operations such as CREATE TABLE and, perhaps more importantly, DROP TABLE. Even indexing and table-restructuring operations are provisional and capable of being rolled back. This is a great comfort and convenience for the DBA, but of course if one prefers not to have this ability it can always be turned off – and then on again, if you like.

Ingres has always had a reputation for technical excellence and innovation, but it also deserves to be congratulated for its compliance with relevant standards. If one downloads the community edition of Ingres Database 9.3 today, the first thing one notices is the wide range of operating systems and platforms supported, including of course Windows, Linux, Solaris, SCO Unix, and even OpenVMS, on both 32- and 64-bit platforms. An obvious consequence of this is that you will run the same software on your laptop as you will on a Sun M9000. There just is no difference in feature sets or syntax. The same enterprise features (such as table partitioning say) that are essential for hosting a 20Tb database on the M9000 can be used on a bottom-of-the-range laptop. For the purpose of testing and development, or even just training, it is a great advantage to be confident that what worked on your desktop will work in exactly the same way on the production hardware.

Each user session runs as a separate thread in the server

Ingres SQL deviates very little from standard SQL. If ISO/ANSI SQL specifies syntax for a particular operation then that is the syntax Ingres uses. I won't go into detail about it here because space is limited, but if you want to know more, Rick van der Lans new book, called *The SQL Guide to Ingres*, is expected to be published in April.

Not only does Ingres support standard syntax, it implements standard semantics too. It is surprising that there is considerable variation in the behaviour of SQL between even mainstream commercial SQL DBMSs, so it is good to know that Ingres behaves in a highly predictable way.

It should also be noted that there are several behaviours mandated by the standards that are actually not particularly helpful to most users. Uniquely for an open source DBMS, Ingres allows you to choose the level of compliance when installing it, by offering the option of strict SQL92 compliance. SQL92 compliance doesn't mean what you think! Unless you know what it means and actually need the surprising and usually unhelpful consequences, don't feel tempted to select it when installing Ingres; it is not easily reversible. However the existence of the option is typical of the way Ingres attempts to fully comply with standards. You can rely on Ingres SQL to behave in a sane and useful way; there are no quirks; no inexplicable ways-of-doing-things that just have to be memorized, and no surprises that aren't inherently part of SQL.

Like all products, Ingres also has non-standard SQL extensions for the things the standards don't address, but even then, Ingres generally attempts to support the *de facto* standard when there is one.

I won't list off a tedious catalogue of every feature that Ingres offers. It's a very long list and you ought to verify it for yourself, either by visiting the Ingres website, or by downloading it and trying it. It *is* very complete.

## Commitment

It's all well and good knowing that Ingres is a viable DBMS that gets the job done today, but that is not enough. You want to know that it has a future and a vision; that the company is strong and growing; that alliances are being formed, and that there are third-parties and a community to turn to for help and advice.

Ingres is offering the option of strict SQL92 compliance

Some of you might even want reassurances about Ingres Corporation's commitment to the open source ethos and community. Any time a previously closed-source product is released as open source it always pays to ask hard questions about the vendor's commitment to open source. Do they *really* believe in the aims of open source and the entire open source culture? Ingres' commitment to open source is beyond question. They did not simply leave the source code lying around on http://code.ingres.com just to fulfil the legal obligation imposed by their GPL 2 licence. On the contrary, they have set up a complete infrastructure and have provided substantial material support. See for example lxr.ingres.com and bug.ingres.com (both of which are Ingres-powered, neatly demonstrating Ingres' compliance with standards and its driver-support).

I hope I have clearly established Ingres' place in the open source world. There are lots of small-footprint engines that give you SQL access to tabular content, and there is no limit to the number of applications that find them sufficient and useful. Trying to use an enterprise DBMS for such applications would be a foolish mistake. A DBMS like Ingres is the right choice when you start seeing requirements for things like role separation for regulatory compliance and full audit capabilities, or robust XA transaction support. Or the need for on-line parallel backup and parallel restore. Or in-place upgrades and a scalable architecture with extensive platform support. Or massive scalability of data, and the ability to fully exploit the very biggest multi-CPU enterprise servers.

All this technology and community involvement is important of course, but it's not enough by any means; having all the right technology at the right price only gets Ingres onto the shortlist. Once it's on the shortlist, decision makers want demonstrable vendor stability. They need confidence that support and upgrades will *always* be available, pretty much forever. Even having the comfort of adopting a DBMS that adheres to all applicable standards, which would be easy to replace if the need ever arose, they still want to make sure they will never have to. Insurance is nice, but one still tries to avoid accidents. It is good to know, as I wrote at the beginning of this article that Ingres Corporation generated more revenue than any other independent open source company in 2008. Ingres Corporation isn't just paying the bills; it is growing and it is investing heavily in new development, partnerships, acquisitions and in sponsorship.

Finally, you surely understand there must be things that I couldn't write about here, today. All I will say is that my *very* strong advice to anyone contemplating an open source DBMS for a mission critical system in the near future, or for analytics, is to set up a Google Alert (www.google.nl/alerts) to monitor Ingres-related news. That's a hint; I can say no more.

**Roy Hann** is co-founder and Technical director with Rational Commerce Ltd. in London (www.rationalcommerce.com) and Chairman of the UK Ingres User Association (www.ukiua.org.uk).