

Evolution of Enterprise Computing

Article for BCS ITNow, by Dr Geoff Sharman FBCS

What's the biggest use of computers today? Page retrieval via the World Wide Web? No. Google searches (Google claims 3.5 billion searches a day worldwide)? Social media? Big Data analytics? No. It's something much more mundane, but also more fundamental: business transactions running on enterprise systems.

If you doubt this, ask yourself whether you did any of the following today: buy something in a supermarket?; use a cash machine, debit card, credit card, or contactless payment card?; use an app for internet or mobile shopping, banking, etc.?; make a telephone call?; trade shares? buy a lottery ticket?; travel by public transport?; watch catch-up or pay-for TV?; use electricity, gas or water? Of course you did! All of these activities generate online transactions which are processed through traditional and not-so-traditional enterprise systems for Online Transaction Processing (OLTP). The worldwide daily volume of these transactions had already topped 20 billion worldwide by the mid 1990s; today it's well in excess of 100 billion. That's two to three orders of magnitude greater than the number of Google searches per day, and equivalent to 14 transactions per day for every man, woman and child on the planet.

All this is simply a reflection of the fact that trade is the dominant human activity: one US bank reports that the financial value of transactions handled by its mainframe-based OLTP system is over \$10 trillion per day. That picture is repeated across government, the financial services industry, other major industries, and web based giants such as Amazon, eBay and PayPal. From the consumer's point of view, commerce has become much easier but the records of our daily transactions are held in systems right across the globe, constituting the "digital footprints" through which our lives can potentially be tracked day by day and hour by hour.

Yet it still comes as a surprise to many IT professionals that OLTP is the major use of computer systems. To understand why, it helps to take a historical perspective. The first use of writing, on 6,000 year old Sumerian clay tablets, turned out to be for accounting records rather than literary works. The first business use of computers in 1952, by the J Lyons company in the UK, was also for accounting, and things didn't change until the US government got involved. In the early 1960s, the US Air Force unveiled its SAGE (Semi-Automatic Ground Environment) air defence system, a very large system combining computers and networks for the first time. American Airlines adapted some of this technology for its SABRE airline reservation system, launched in 1965. That year also saw other key developments in computing: IBM's introduction of System/360 - the first series of upward compatible computers - and the formulation of Moore's Law. Which of these had the greatest influence on subsequent events?

IBM quickly moved to become the largest computer company of the era. The core of its success was the widespread adoption of IBM software, such as BATS (Basic Additional Teleprocessing Support) implemented for UK banks, together with database management systems and communications software based on SNA (Systems Network Architecture) as the basis for enterprise systems. Within a decade, these systems were installed in large enterprises across the globe. Many of those systems have been continuously enhanced and remain in use to the present day, forming a large part of the "digital infrastructure" which we all now depend on.

Gordon Moore, the founder of intel, announced his famous "law": a prediction that the number of transistors per unit area of silicon would double approximately every two years for the foreseeable future. Over the next forty years, this exponential increase in processor power at constant cost changed everything: cost/instruction, market size, device form factors and network bandwidth, opening up the era of "pervasive computing". Few computer companies of that era survive to the present day. How is, then, that OLTP survives and thrives?

To answer this question, we first need to understand what it is. The term “OnLine TeleProcessing” was initially used to denote the use of terminals connected to a central computer via telephone lines, and only later adapted to encompass the meaning of handling business transactions with reliability and integrity. These terminals were used by employees of airlines, travel companies, utility companies, and banks to capture customer transactions at source and process them immediately, rather than filing for later action. Consumers usually had no access to these terminals, although banks were the first to offer consumer terminals, e.g. the Lloyds Bank Cashpoint (IBM 2780) which was introduced in 1972.

Typical networks of the time were small (~50 terminals), but a key problem was handling concurrent activities efficiently; network lines had low bandwidth (~1024 bits/sec) and could not be shared by different applications; processor hardware was slow (~1 MIPS); while accessing data (typically held on magnetic tape) would have been even slower (seconds/record). Hardware advances including higher speed leased lines, direct access disk storage, and the continued improvement in processor speed were important enablers for these systems, but even they could not match the market demand nor keep pace with its rate of growth. That required a radical re-think of the way software was designed.

The key problem was that the operating systems, data management systems and programming languages of the day had all been designed for batch processing, where a single application program might run for hours. Scheduling a batch job (process) could take millions of instructions, i.e. many seconds. Most operating systems could only handle a few concurrent jobs; data management mainly provided support for sequential files; while programming languages did not support network operations. None of this technology fitted an environment where each terminal user needed a response within a couple of seconds.

It quickly became clear that new software was needed for the management of indexed files and data bases, which allowed direct access to a specific record or sets of records within a data file in milliseconds, for data communication, which enabled receiving and sending messages and control of telephone lines, and especially for application execution which enabled rapid scheduling of short application segments. These segments are initiated by a message from a terminal and create a response message in real time; the operating software became known as an OLTP Monitor. Some well known monitors first released in the late 1960s and early 1970s have been continuously enhanced ever since and one of these, CICS (Customer Information Control System), has been described as “the most successful software product of all time”. It’s been developed at IBM’s Hursley Lab in the UK since 1974.

At the heart of OLTP is a programming model where resources such as applications, memory, processes, threads, files, databases, communications channels, etc. are owned by the monitor rather than by individual applications. On receipt of a request message, the monitor initiates an application segment and provides concurrent access to these resources. It frees resources when a response message is sent, so the application has no memory of actions it may have performed or, in other words, is “stateless”. More complex applications (known as “pseudo-conversations”) can be created by retaining some state data in a “scratchpad area”. The next segment retrieves this data from the scratchpad area, processes the incoming message and issues another response message. This model is very different from conversational application models, which retain large amounts of state between user interactions, and it’s this feature which enables it to scale to thousands or millions of transactions per second.

Subsequent decades saw intense competition for this OLTP business from vendors of mainframe compatible systems, specialist “non-stop” systems, mid-range and Unix systems, packaged application systems, and PC-based distributed systems using local area networks. The most far reaching impact, however, was occasioned by introduction of the World Wide Web as a service running on the Internet in the early 1990s. The WWW used a stateless activity model in which messages triggered the retrieval of a web page and generated a response message. It became immediately

apparent to many observers that this was, in effect, a read-only version of OLTP and it wasn't long before creative programmers found a way to run user applications from a web server. The original WWW pioneers, by contrast, knew nothing about enterprise applications.

Opinions varied on what impact the WWW would have on existing OLTP systems. Some commentators thought it would provide the low cost global network which these systems needed, while others thought it would spark a wave of new products which might compete with those systems. In the event, both have happened. Rapid innovation has seen the creation of new web-based programming models and languages such as Java, PHP (used as the basis for Facebook), and Node.js (a variant of Javascript), each of which implement aspects of the OLTP programming model. At the same time, established OLTP monitors have incorporated support for Internet protocols and new programming languages, and now sit behind many popular web sites for Internet banking, shopping, travel reservations, and so on. One mainframe based travel reservation service processes a billion transactions/day on its own.

There's no doubt that OLTP, in its many forms, powers the world economy and will continue to do so. To take only one example, sensors connected to the WWW for the Internet of Things will require OLTP applications to support them and are likely to drive the global transaction rate to trillions per day.

For more background on this article, see <http://www.bcs.org/content/conWebDoc/54889>