

# White Paper



## The growth in data volumes

...an opportunity for IT-based analytics with Pervasive DataRush

DataRush represents a significant opportunity for many organisations faced with analytics applications that simply can't deliver answers fast enough

David Norfolk

## Executive summary

We currently see a real and emerging set of data-related and business-oriented issues facing corporate IT:

- An increasing focus on corporate governance, which requires flexible and near-real-time analytics, often compromised (up to now) by the necessary data cleansing and transformation overheads associated with processing huge volumes of data;
- The difficulties of programming for the new parallel multi-core computer architectures (which are being forced on organisations for perfectly reasonable, computer-science-based reasons, by all computer vendors) and the associated people/skills issues;
- The need for cost-efficient (and demonstrable) utilisation of all available computing resources in an organisation as the recession bites, as well as the usual focus on keeping operating costs down; and now coupled with a need to address the green issues of power consumption and power grid availability.

A fundamental underlying issue, however, is that the volume of data stored by businesses as a result of normal day-to-day processing is increasing hugely and this represents both a significant investment and an opportunity for IT to use analytics to support fact-based decision making in the business; and, incidentally, to demonstrate its positive contribution to delivering the business outcomes that the business needs. This situation is largely the result of 4 factors:

1. An increased focus on the use of automated business service delivery to enable business outcomes (as evidenced by, for example, ITILv3); and, therefore, a renewed focus on IT as an enabler of the overall organisational strategy;
2. A rapidly reducing cost of storage coupled with rapidly increasing storage densities, which together remove some of the physical barriers to storing large amounts of data;
3. An increasing acceptance of the need for the continuous capture of all possible digital data and changes to digital data by default (this is partly driven by evolving compliance and audit requirements);

4. The plunging cost of computing power as energy-efficient multi-core architectures become the norm.

However, although the explosion of data in the business is an uncontroversial observation, it is far from clear that this data is being exploited effectively in many businesses. In fact, DMReview magazine has reported BI (Business Intelligence) surveys showing that the median volumes of data in BI applications haven't changed much over the last 5 years, at a mere 5–6 Gb.

This means that it is important to distinguish data (of which we have vast amounts) from information (data which is understood) and knowledge (information which is used by the business). Many data-rich businesses are both information- and knowledge-poor and thus are missing the revenue opportunities that come from making that transition—they are, in effect, wasting resources. This isn't just a commercial issue; something similar affects Government organisations, of course. They collect vast amounts of data during their day-to-day operations and have an obligation to exploit it; not for revenue so much as for the provision of security and services to their citizens.

There are many possible reasons for the limited exploitation of data resources in practice. It is possible that there is actually little value in all this data (in which case, why waste resources collecting it) and, even if it is valuable, that all the value available and relevant to any particular question can be extracted from well-defined subsets of it. In some cases this may well be true; although it seems rather unlikely that it's always true and hard to understand how people could ever be sure of this. Besides, in order to prepare a meaningful subset of a given kind of data for processing, an organisation should select from all of it, so it still has to cope with the overheads of processing (cleansing, say) large datasets in their entirety and selecting important records.

## Executive summary

---

More likely, perhaps, is that the conventional technology available to many business organisations can't cope with processing the volumes of data now available to them; it is also possible that the data as a whole isn't well-enough understood (or high-enough quality) to make the use of more than defined subsets of it feasible. There is then the real risk that the data subset used for supporting a particular decision or governance report is biased because only the data the organisation can afford to process (limited by time and resource factors) is sampled, resulting in poor quality business decisions or reports. The value and risk associated with using a data subset will depend on the potential cost of a bad decision or poor reporting information, of course, as well as the scope of the decision.

Surprisingly, it is also possible that some organisations aren't really prepared to make fact-based decisions at the level of detail implied by the analytic processing of any or all the data in the organisation on a routine basis. The use of business analytics, in theory, is accepted without question; but there is often real resistance in practice—decision-makers have power, which (in dysfunctional or immature organisations) can be reduced by the transparency of an analytics-based, fact-based, decision making process. Analytics are one aspect of BI in general and noted BI expert Mark Whitehorn (in his article *The Business Intelligence (BI) scandal: why pay more to get less?* for *The Register*) says: "...there are plenty of coalface jobs where access to a BI system could make a huge difference. BI systems can detect real-time credit fraud; they can warn sales staff of potentially abusive or dangerous customers, it can alert shelf stackers to changing shopping requirements and so on. However, I also believe that deployment at this level is far more likely to be hindered by human/social reasons rather than technological ones."

Nevertheless, we believe that the vast amount of data now available to organisations does, in fact, represent a valuable resource if it can be mined and exploited in a timely way by organisations prepared to move up the organisational maturity scale. However, such organisations will then meet significant technical obstacles to achieving the business outcomes they want:

1. Processing of very large amounts of data in conventional ways, even just to select a subset, is slow and expensive;
2. Most data needs cleansing of spurious, duplicate, inaccurate or incomplete records before it can be used and this often takes longer than the analytics processing itself;
3. High-speed specialised computing power is very expensive and reliably programming the new generation of powerful commodity multi-core computers is very difficult;
4. Thinking parallel needs a change in programmer mindset and many organisations have chosen programmers for their ability to take advantage of ever faster single core and single threaded architectures. Making these people code parallel systems at the detailed level, to process large datasets efficiently, would mean managing disruptive cultural change.

Pervasive's DataRush, which is described in more detail in this paper, is one new technology that could address all these issues; by enabling efficient utilisation of cheap, multi-core, commodity computers for near-real-time cleansing, ETL (extract, transform and load) and processing of very large data volumes. It deals with the skills issues of programming for multi-core systems by dealing with parallel programming in an underlying framework, which presents a simple and familiar Java interface to an organisation's programmers, who don't need to worry about the underlying process.

What DataRush delivers to the business is the ability to exploit the rich data resources it owns (typically, but not exclusively, for governance and decision support applications), with demonstrable cost-efficiency, unhindered by conventional IT processing bottlenecks.

## Executive summary

---

### Fast facts

There are, we believe, four major pre-requisites for the exploitation of large volumes of data with BI analytics:

1. The availability and use of technology to extract, store and manage the metadata (semantics) associated with company metadata, so that the business owners of the data can find out what is available to them, in business terms;
2. The availability and use of technology to process very large volumes of data efficiently and effectively, in a timely manner (we go into some of the issues around this in the body of this paper);
3. The maturity to set a baseline so that the value of the analytics can be demonstrated (thus avoiding the risk of analytics for their own sake);
4. The maturity to accept a fact-based decision making process.

Pervasive is one good example of a company that supplies technologies that can help satisfy the first two of these prerequisites.

### The bottom line

The very large data collections available as a by-product of today's automated business processing do represent a huge potential opportunity, we think. The goal is fact-based decision-making for everyone, which sounds very attractive—but, as usual, the devil is in the detail.

Exploiting this opportunity needs both appropriate technology—something that could be called Data Intensive High Performance Computing technology—and a real degree of organisational maturity. Pervasive's DataRush technology is a good example of an enabling technology in this space for data-intensive applications, but it is no free lunch. DataRush is only appropriate for a subset of applications (data intensive ones) and existing applications need to be modified to take advantage of it (although DataRush does protect programmers

from the detail of parallel programming]. The organisation involved must also want to take advantage of fact-based decision making and associated knowledge-based activities such as predictive analyses, event processing, automated rules generation (i.e. self learning algorithms), identity assignment/management, and fraud detection.

That said, DataRush represents a significant opportunity for many organisations faced with analytics applications that simply can't deliver answers fast enough to satisfy the business' needs using conventional technology. It has general applications for data-intensive processing but we think that its initial adoption will be driven by use cases where alternative approaches are infeasibly slow—which then makes DataRush the lowest risk choice. A business process that works when information is delivered to the user in seconds (a sales process that analyses a customer history to suggest products the customer might like, perhaps) may simply not work when information is delivered in minutes or hours—and DataRush has demonstrated orders of magnitude speed increases on real-world data analysis problems.

## Growing data volumes—opportunity or curse?

### Data growth and the rise of data-intensive high performance computing

In our opinion, the nature of the business-oriented IT industry is changing as businesses start to focus on automated holistic business service delivery (the rise of ITILv3 is both a symptom of this changing focus and its enabler). A holistic business service delivers a complete business outcome at some level, from a mixture of automated and (usually) manual activities; is specified in business terms; and has a service level (formal or informal), again specified in business terms. A customer database with update transactions taking under .5 second is not a business service (but it may enable one); a customer management system capable of acquiring, updating or removing up to 5000 customers per day, coping with peaks of 10x normal levels, and providing customers with a self-provisioning service via the Internet might be.

One side effect of this change is increasing interest in IT as an enabler of the overall organisational strategy—in other words, as the supplier of automated analytics to support fact-based decision making in line with the organisation's strategic directions. In the past, analytics have been reserved for specific high-value decisions but as the tools improve there is pressure (not least from BI product vendors) to extend this to decisions throughout the organisation. Information Builders, for example, calls this Operational BI (Business Intelligence) for the Masses; Calum Nobles, technical director for EMEA at Information Builders says of BI in many companies: "They're still talking about executives, planners, big power users... The bit they miss out is getting the plan executed [throughout the company] and that is what operational BI is all about, giving the operational users the measures, the KPIs [Key Performance indicators] they stand or fall by". Obviously, key to getting fact based decision-making out to the masses is getting the right data sets into your automated analytics processing in a timely manner. Nevertheless, capturing, managing and processing potentially very large volumes of data isn't easy.

However, the physical barriers to storing very large amounts of data are decreasing. This isn't the place to go into storage technology, but when you can buy commodity half-terabyte hard drives at consumer prices from your local supermarket this is a good indicator that cost and availability of storage isn't a barrier any more. This has changed user perceptions: people no longer see running out of disk space as a problem. This changing perception has been helped along by regulatory pressures. Sarbanes-Oxley, for example, was often (misrepresented) as saying "keep everything for ever" and, although the UK Data Protection Act also says "throw personal data away securely when you no longer need it", the feeling has grown up that keeping transaction logs etc. for a reasonably long time is a good idea, in case some regulator asks for them. After all, the UK government even appears to have plans for a super-database containing the details of all phone calls and e-mails sent in the UK.

Less sensationally, MiFID requires financial organisations to keep logs of financial transactions for several years, so as to enable verification of historical best-pricing. Providing transparency back into past business transactions is the direction future regulation and governance is going in. In fact, an August 2008 Deloitte poll is reported as saying that "two out of five executives say company data volume is increasing in size and becoming unmanageable"; according to Bruce Hartley, a director in Deloitte's Analytic and Forensic Technology (AFT) practice, "As the volume of data continues to amass—doubling in size every 18 to 24 months—strategic steps should be taken so that electronic discovery can be handled correctly".

However, we should also make the point that keeping data beyond the point when you have a business use, or a regulatory requirement, for it is not a good idea (if you are ever the subject of litigation, for example, why give expensive lawyers more data to sift through, while charging you for doing so). Nevertheless, remember that discarding just the right data, securely, implies that you can process however much of it there is in a timely way.

## Growing data volumes—opportunity or curse?

So, we have increasing volumes of data available for analytic BI (and other) processing. The next issue is whether we need to process all of it, all the time, in order to get useful results. Clearly, we do not. Statistical sampling can give acceptable confidence limits from processing just a properly chosen subset of the data.

Unfortunately, “properly chosen” is the key phrase here—10% of your total database may be sufficient, statistically, to answer a given question reliably (there are formal statistical methods for determining sample size), but not if the selection process is skewed by, for example, leaving out records that aren’t available locally or you haven’t got around to processing (validating) yet. Potentially, you need to be able to process all of your data, in a timely manner, in cases where sampling introduces risk or a missed event could be very expensive. Also, of course, any sampling algorithm must have access to all the data in a cleaned-up (accurate, de-duplicated) form as, otherwise, its samples may be skewed. This means processing all of your data before you extract your sample and, with conventional technology, this can take longer than the analytics processing (which can’t even start until cleansing is finished).

This can be considered an emerging High Performance Computing (HPC) problem—the ability to process very large volumes of data in near real time. Latency, the unavoidable delay between asking a question and getting the answer, is a business issue in the Internet-connected world and, in general, the business demands lower latency. High Performance is usually taken to mean high performance making calculations (which it still is, partly) but high performance in terms of data throughput (to and from very large (terabyte scale) data stores) is also a critical aspect of performance for many businesses.

Luckily, the cost of high performance computing is plunging as energy-efficient multi-core architectures become the norm. However, in order to exploit this power, technology needs to be able to take advantage of parallel processing on multi-core architectures; and this turns out not to be trivial.

Looking forward, there are 4 big issues that data-intensive HPC tools will have to address if they are to be useful to any organisation with a need to process very large amounts of data in the shortest possible time and at reasonable cost. Addressing these issues underlies the design of Pervasive’s DataRush framework, which will be considered later in this paper. These issues are:

1. Coping with the technical implications of processing very large datasets, as well as handling computationally-intensive workloads at the same time. Memory is cheap and available, but there still comes a point where everything won’t fit in memory. Dataflow approaches (as used in DataRush, see below) are inherently frugal with memory, as parallel subsets of the data stream off disk for processing and back onto disk again.
2. Coping with ubiquitous parallel computing. Both the CTO of Intel and the Chief Scientist of Microsoft have described the multi-core revolution as the biggest deal to hit computing in 30 years and multi-core is no longer the preserve of the supercomputing high end. This is coming about mainly because energy-efficient parallel computing power, as it scales up, is increasingly cheaper than ever-faster single core approaches (in the extreme, just the electrical power needed by data centres, both for actual processing and for cooling the processors, is becoming a barrier to growth). The trouble is, that programming for parallel computing is hard and it is very easy to write applications that will run on multi-core processors but only use a fraction of the cores available (and Intel is already thinking in terms of hundreds of cores, not of 4 or 8 way systems). An important way of addressing this is through high-level frameworks which inherently support parallel programming on multi-core computers, transparently to the programmers.

## Growing data volumes—opportunity or curse?

3. Coping with cognitive load and programmer productivity. Traditional methods of programming for parallel computing need sophisticated highly trained programmers and, even then, not everyone can think parallel effectively. Commentator Phil Manchester has reported in The Register on how Intel is getting together with representatives from AMD, Nvidia, Sun Microsystems, academia and the open source movement to address the parallel programming skills gap and find ways for the industry to get universities to break with their attachment to traditional sequential programming. Even if you can find (and afford to employ) specialist programmers for the parts of your systems that must take advantage of many parallel processors (and, as we've seen, that will probably be all of them in the near future), conventional low-level parallel programming, and the system design behind it, is shockingly unproductive. One approach to this issue is to use parallel processing frameworks (using innovative techniques such as dataflow programming) to enable developers to programme massively parallel data-intensive applications without having to worry about the hard parts such as memory management, threading, queueing and deadlocks.
4. Coping with economics. Since energy efficiency is one of the main drivers for multi-core, and we are entering a period of recession, inefficiently wasting cores and the cheap processing power they represent is obviously unacceptable. For the foreseeable future, the requirement is to do more with less, whether this is dressed up as green computing or dressed down as hard-boiled economic reality. Being associated with waste in any form will, potentially, damage an organisation's reputation and even its share value. This calls into question conventional cluster-based, distributed-parallelism solutions to scaling up performance as these are based on separate but linked computers, associated with ever-increasing management and software-licensing overheads and underutilisation. Next-generation, multi-core friendly, fine-grained, massively parallel data-processing engines, which can take advantage of commodity symmetric multiprogramming (SMP) boxes, may well bring a compelling price/performance advantage.

### Data-intensive HPC: good practices

In order to exploit parallel processing architectures, it is desirable to employ modern programming good practice generally. Component-based software, with low coupling between small, cohesive components, will usually parallelise well. Avoid global, updatable variables, as update locks can force serial processing. It is usually best to code using frameworks designed for parallelisation rather than trying to code everything from scratch, so that general and (especially) maintenance programmers are isolated from the technical details of parallel processing.

There are several desirable characteristics for any business-oriented approach to achieving parallel processing:

1. Ideally, it shouldn't require any changes to existing programs. This is, in general, not achievable (a compiler, for example, can't automate parallelisation because it doesn't know about the run-time environment); even compiler hints represent code changes, which must be tested. So, program changes can only be minimised.
2. It shouldn't require programmers to learn new programming languages.
3. It shouldn't introduce unexpected side effects. In general, this isn't achievable as race conditions and deadlocks are easier to achieve on multiple processors and often far from intuitive; but a suitable framework can help minimise these effects.
4. It should allow programmers to think serially and write applications which still run parallel—programmers shouldn't have to worry about the details of scheduling instructions around different parallel processors.

## Growing data volumes—opportunity or curse?

### Related issues

Data intensive HPC applications will also have to deal with three issues common to most analytics applications:

1. The data integration issue: how do you efficiently (quickly) connect to disparate data sources, often with different and incompatible formats?
2. How do you cope with data quality issues: often, an organisation only finds out how poor quality (inaccurate and incomplete) its data is when it tries to use all of it at the enterprise level and then gets bogged down with the cleaning up of the data before it can be used.
3. The movement issue: moving data around is slow; you want to process the data you have as and where it is now, you don't want to waste time moving it somewhere else and reformatting it.

It is important that any data-intensive HPC tools either help you to address these issues or link to other tools that can help.

### The technology vendor landscape

Most BI software vendors are endeavouring to address the multi-core issue, and Intel (obviously) is actively addressing multi-core issues generally. Multi-core processor architectures are unavoidable in future, as they address emerging energy efficiency and scalability issues.

The problem is that the computer industry has relied for too long on serial Von Neumann architectures, with a single processor

that processes instructions in strict time order. These, in general, can't be generalised to multi-core architectures where several things can happen at once, in indeterminate order (if one process takes longer than expected, without extra controls—known as locks and semaphores—a subsequent process may start on a different processor and complete before its strict predecessor has finished). Even Intel's compiler-oriented solutions need programmers that understand multiprocessing issues, although it supplies tools that help such people visualise what is going on in complex parallel-processing systems. Sharing workloads across clusters of these serial computers is a partial solution to increasing processing power in the short term but it tends to be expensive and wasteful at the individual computer level (not least, because of the way most software is licensed currently, although this could change).

The obvious solutions to the data-intensive HPC issue come from the appliance vendors, Netezza, Azul and the like, as their appliances are highly-parallel processors themselves. However, indexing specialists such as CopperEye and database vendors such as Sybase (with Sybase IQ) and Pervasive (with DataRush) can supply high-level technology which copes well with the data-intensive HPC subset of the issue; and which is, of course, itself entirely suitable for deployment on highly-parallel data-processing appliances.

So, technology is available to help you turn growing data volumes from a curse to an opportunity. Whether you can take advantage of it depends on whether you have a clear vision of the possibilities for fact-based decision making for everybody in your organisation, of course.

## Pervasive DataRush

Which brings us to Pervasive's contribution to the data-intensive HPC landscape: the DataRush framework. Given good metadata management (that is, that you understand the structure and semantics of your data well) DataRush avoids the need to move data into structured storage—it can be processed and analysed as-is (in transaction logs, for example). You can extract statistically significant unbiased subsets of all of your data or you can avoid sampling issues by processing the whole lot.

DataRush targets large, batch-oriented, long-running database applications—of the sort that are currently threatening to deliver information only after the need for it has passed, as data volumes move past the terabyte barrier—that must now run on multi-core platforms. It also protects programmers from the details of parallel programming.

The DataRush approach is based on a technique called dataflow processing, which delivers parallel processing of subsets of the data, divided up based on an understanding of its properties. Data analysts or database specialists generally understand this approach with little training; for general business programmers, this may take only a little longer. Nevertheless, don't think of DataRush as a free lunch; programmers will need to understand a new approach to programming. Although, DataRush does at least use Java, so it doesn't require programmers to learn their way about an entirely new language and environment.

DataRush manages parallelisation using Java components (known as operators), which are either chosen from a large library or written by the organisation processing the data. It can dynamically take advantage of information such as the number of processors that will be available at run time, in order to partition the data into independent units (which can be processed in parallel) and to control processing flow generally. These dataflow techniques are useful mainly in what are traditionally called batch-oriented data analytics applications, but where they are useful, they can speed up the delivery of results by an order of magnitude or more.

The dataflow technique of computer processing it uses is based on Kahn networks and Parks scheduling and isn't new to computer scientists, although its application to general commercial programming is novel. It involves a new way of thinking about pipelined applications, a new process of thinking about the data

operations; and remember that programmers generally don't like change (or, at least, don't like being changed without consultation).

Nevertheless, DataRush doesn't make you throw away your existing programmes entirely and rewrite them from scratch, although it doesn't need just a simple recompile or automated port either. You do need to rewrite some of your code and you probably need good, professional, programmers who are parallel-aware (although not necessarily parallel-programming specialists) who will follow established good practice programming techniques. Even if a framework supports parallel processing, bad coding techniques (such as a single global counter, locked for update by each process in turn) can probably force programs using the framework to run serially.

### DataRush implementation

DataRush is a framework that sits on top of a Java Virtual Machine (JVM), which itself runs on the usual Java-supporting hardware from Sun, IBM, HP, Dell, Azul and so on. The framework is operating system agnostic; it will run on Linux, Solaris, AIX, HP UX, and Windows Server. Nevertheless, the efficient use of multi-core hardware by DataRush can be affected by how well the JVM underneath has been implemented.

The DataRush framework provides classes for, for example, configuring a DataRush application, possibly using configuration information from outside the application; and for partitioning and defining dataflows. It provides operators for partitioning and de-partitioning flows of data, for reading from JDBC sources and writing to JDBC targets, among other tasks. It also provides tools to assist with testing DataRush applications and provides generally useful utility programs. It covers the process from Extract, Transform and Load (ETL—to get data into a system) through profiling the data and running the processing logic to the output of the results of the processing. The framework abstracts the mapping from a DataRush analytics application to the underlying multi-core hardware by means of a dataflow programming model, so that programmers are largely insulated from the issues of scheduling and locking threads running on parallel processors.

In addition, DataRush comes with packaged applications for, for example, fuzzy logic data matching and more of these are being written.

## Pervasive DataRush

Pervasive DataRush has recently been made available for general release and can be downloaded today. Also available is detailed JavaDoc documentation, which goes into the technology more deeply than we've done here.

However, the core DataRush technology is used in Pervasive Data Profiler, which has been a shipping product for some time—DataRush technology is reasonably mature on general release.

### Product architecture

Pervasive DataRush is a Java framework. This means that it abstracts (virtualises) the underlying Java Virtual Machine (JVM), which, in turn, virtualises the underlying hardware. The threading and synchronisation involved in parallel processing can be handled entirely by the framework because data is only shared at a high level through inputs and outputs. A DataRush operator (which does something useful at the user level using DataRush technology) is an extension of an existing class in the framework and a library of useful user-level operators provided as part of the DataRush framework as well as the dataflow engine that constructs and executes dataflow graphs in Java and the low-level routines it uses. Further user-level operators can be written in-house, if needed. See Figure 1, The DataRush Architecture.

### Differentiators

DataRush's key differentiator, at a business level, is that it can reduce processing time for an existing business application from hours to minutes or even seconds. This can make the difference between an automated business process being feasible and it being infeasible, as data volumes increase.

It does this by flexibly making use of however many CPUs are available to it at run time—if you need faster processing, simply make more CPUs available, with no need to reconfigure or redesign the program. This delivers linear scalability from commodity multi-core PCs according to Pervasive's own tests on a publically-available benchmark: see Figure 2, DataRush Linear Scalability, which shows near-linear scalability up to 8 cores.

In addition, the abstraction layer provided by the DataRush framework largely allows programmers to ignore parallel processing issues (number 4 in our list of desirable characteristics of a business-oriented approach to achieving parallel processing, above). It should also materially assist with avoiding the undesirable side effects of parallelisation (desirable characteristic 3), as long as DataRush programmers use accepted good practice. Specialist languages already provide similar abstractions, but DataRush allows programmers to

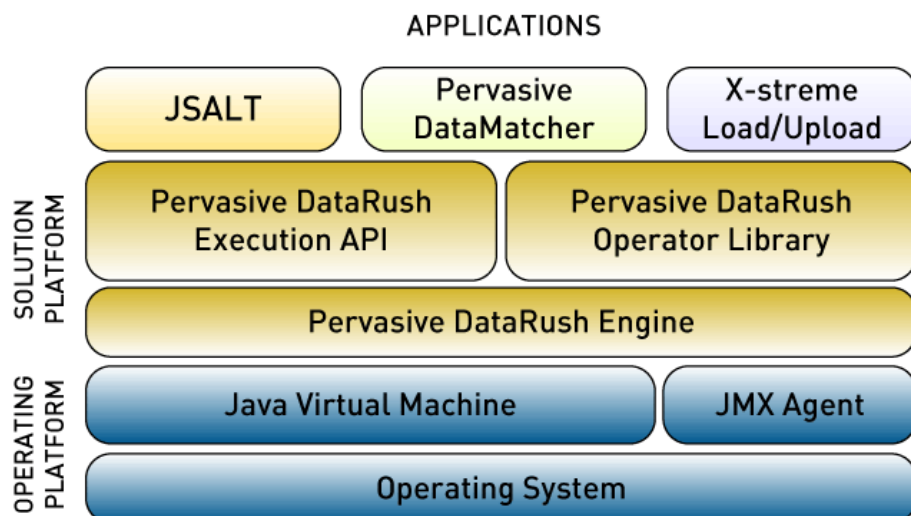


Figure 1: The DataRush architecture

## Pervasive DataRush

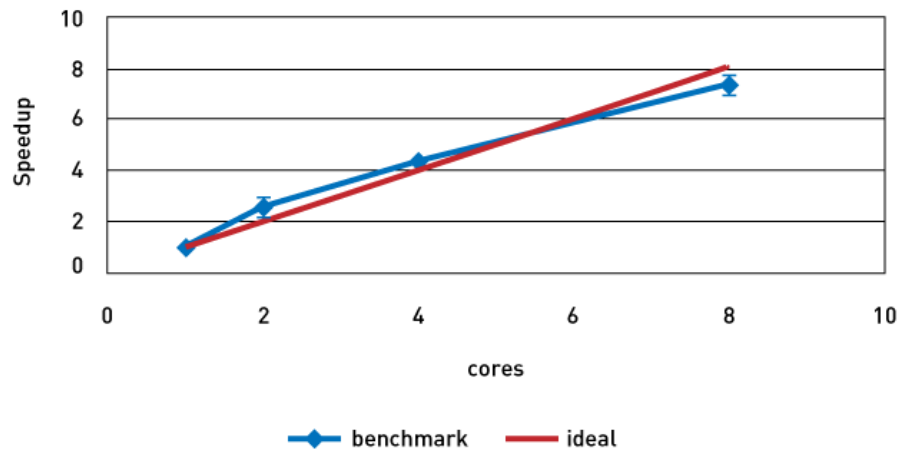


Figure 2: DataRush linear scalability

work in a Java environment that they are familiar with (number 2 in the list of desirable characteristics).

The DataRush implementation minimises the need to make changes to existing programs that are being migrated to the platform (number 1 of the desirable characteristics) but it can't eliminate this. DataRush applications must be written by skilled programmers who know what they are doing.

DataRush also comes from a company with extensive experience in data integration, which helps it to deal with the Related Issues noted above: connecting to disparate data sources efficiently and data quality. Its dataflow approach and ability to cope with comparatively raw data helps it to deal with the related issue of a need to minimise data movement too.

### Supporting products

DataRush complements Pervasive's other data management and data integration technologies (such as Pervasive SQL) but is largely independent of them.

However, absolutely key to effective analytics of all kinds, as well as data integration, is managing data quality and here Pervasive provides a solution called Pervasive Data Profiler, which is built using DataRush technology and is a shipping product. This, together with Pervasive's 20 years experience in data integration, helps to address the data-intensive HPC

related issues of data integration and data quality, mentioned above.

Pervasive Data Profiler gives a good confirmation of the capabilities of the DataRush framework in practice: parallel execution with automatic sealing across multiple CPUs.

Pervasive Data Integrator provides a comprehensive data integration software solution that can be used to orchestrate DataRush applications as part of a bigger system.

The latest DataRush-based tool, Pervasive DataMatcher, addresses the unexpectedly demanding and very common need to match data across large disparate datasets. As data volumes increase, this rapidly becomes a major resource hog: comparing every record of a 100,000-row dataset involves nearly 5 billion individual record comparisons. DataMatcher is designed to match on a subset of the fields in a dataset, or all of them, and includes fuzzy logic matching—and it uses the DataRush architecture to enable just about linear scaling as more processors are added on multi-core devices.

### Use cases

Pervasive itself recognises that DataRush isn't suitable for every data-intensive application but it is useful for a lot of them. Its sweet spot (but not its only application) is for processing large amounts of raw row and column data without going through the overhead of converting it to a more structured format—and

## Pervasive DataRush

---

without the necessity of moving it into a relational database, with the time delays that implies for a large dataset. Suitable applications for DataRush are best illustrated with real customer examples.

So, a data outsourcer uses DataRush for pre-processing structured flat-file data for health insurance audit applications. Its claims management application had been successfully analysing up to about half a million claims so effectively, looking for billing issues, that its customer wanted to scale up the application to processing files containing tens of millions of claims. Unfortunately, using conventional technology, the time for pre-processing that size of file was estimated at around 26 days. However, by invoking Pervasive DataRush with a process orchestrated in Pervasive Data Integrator (using all the cores available in an AMD Barcelona 16-core 64-bit server containing four quad-core processors and running Windows Server 2008), the application was able to process 250 million claims in less than one day. Now, the data outsourcer wants to offer an HIPAA compliance application using Pervasive DataRush's abstraction layer to segregate patient-specific information for a third-party hosted application, where both security and scalability will be a vital part of the offering.

Moving up a notch in sophistication, another data outsourcer in financial services uses DataRush in a loan-tracking and validation application, back-ended to a SQL database. It analyses structured flat-files from hundreds of disparate data sources, involving hundreds of millions of records and, once again, the Pervasive DataRush engine and library was used, embedded in the higher-level Pervasive Data Profiler, running on a commodity multi-core server. From the application developers' point of view, they don't need to work with threading, concurrent memory access, deadlock detection, data workload partitioning/buffering and other complex aspects of parallel thread programming: Pervasive DataRush handles all of that and provides high level components for sorting, collation, Extract Transform and Load (ETL) and so on. It creates individual units of work that are inherently parallelisable and linked into a dataflow graph that can be reused elsewhere (see Jim Falgout, *Crunching Big Data with Java* [One Team, One Month, One JVM] in *JAVA Developer's Journal*. The net result, according to Pervasive's CTO Mike Hoskins, is that: "On

an 8-core machine, the customer processed something like 800,000 loans with nearly 1400 complex metrics, rules and operations per loan, in a runtime of only 3 minutes. Throughput was about 6.6 million data operations per second." However, the real point of the DataRush approach is that the customer expects this to scale automatically with more cores, efficiently utilising all available processors, as more cores become available on affordable commodity machines.

However, DataRush also copes well with conventional structured SQL data-matching applications. A systems integrator in the logistics data-publishing field uses it to support an application that provides current, accurate and comprehensive data on international trade patterns from millions of customer shipping manifests. Pervasive DataRush supports the master data management side of this, using fuzzy logic matching and record linkage algorithms to de-duplicate some nine million master company records collected over 23 years. According to the systems integrator, a key stage in the process involves clustering disjoint sets and an initial approach, built on a mainstream relational database, took three hours to cluster 14 million records. Using the Pervasive DataRush architecture however, and bypassing the database altogether, clustering these 14 million records took just 22 seconds on an 8-core server: parallel processing scaling seamlessly across all 8 cores delivered a 490-fold performance improvement. What this means for the business is greater data accuracy, as data is validated against standards, with extraneous and redundant entries removed; and delivery of the cleansed data to the business more promptly after its arrival. Moreover, fine-tuning these data cleansing functions when the business' requirements evolve merely involves changing configuration files as the schema changes, rather than changing code.

A final use case involves the other end of the DataRush value-add: it claims to enable developers to deliver parallel processing applications without the necessity for extensive retraining or even the employment of new people with new skills. We asked a Pervasive developer, Dr Nena Marin, who has been through DataRush reskilling recently, what she thought the retraining overhead was like in practice.

First, we asked about the skills needed to use a dataflow-based application: "it's as easy as

## Pervasive DataRush

---

editing a properties file to configure it for the current environment”, she said. “The minimum requirement is to be able to edit a properties file and to be able to run a Java application”.

Then we asked about composing a dataflow application by combining high-level operators from the DataRush framework. Here there is a jump in understanding, as the programmer has to start thinking in terms of data streaming in and being processed as it goes; and then being streamed back onto disk. It is no longer all read into memory and processed as a whole, in memory. Once this adjustment is made, programming is “as easy as invoking operators from the DataRush Java class library,” according to Marin, “so the minimum requirement for programmers is that they can compose a Java application based on components/operators provided in a Java class library [DataRush] and described in JavaDocs”. We’d suggest that any prospective customers should undertake a DataRush pilot study, using their own data and an intractable, important, but not mission-critical problem, to reassure themselves on this point.

So, who creates these operators and what skills do they need? “To create DataRush operators,” Marin says, “you must be a proficient Java programmer capable of extending base classes in object oriented fashion using Java”.

## Summary

---

Success with bringing fact-based decision-making to the masses depends on many factors, including access to, potentially, all the data in the organisation in near-real-time and the amount of data in commercial organisations (as well as in government departments) is increasing rapidly. John Bernard, Product Line Manager for DataRush, has likened data collection to a tsunami: "it's happening so fast that both public and private sector organisations simply aren't prepared to deal with the wave".

In practice, of course, most organisations will only use a fraction of this potential data resource, at least for now; and near enough real time may involve latencies of minutes or even hours or days. Nevertheless, it is important that these compromises are driven by business requirements, not the limitations of technology.

So, in general, you should use innovative technology to:

1. Help you manage data quality.
2. Help you associate metadata with these very large data collections and manage this as it changes, so that anybody with a need for analytics can understand the semantics of the very large data collections that are increasingly available and formulate appropriate questions against it.
3. Manage fast access to very large data volumes—so that the data samples chosen for analytics applications are limited by statistics and business need, not the technological problems with accessing large data stores; and so that decision support can be supplied in a timely manner.

Pervasive's DataRush and its associated technologies (such Data Profiler) can help you with all 3 of these. It is particularly appropriate to addressing point 3. However, this is not sufficient for success. In addition, you need to:

4. Understand the desired business outcomes for the organisation—so that you eliminate any tendency towards wasteful analysis for its own sake,
5. Understand and manage the social and organisational barriers to implementation that often discourage the use of fact based decision making in practice.

Technology can assist here too, but more important is the organisational and management maturity of the organisation concerned.

### Further Information

Further information about this subject is available from <http://www.BloorResearch.com/update/1024>

## Bloor Research overview

---

Bloor Research is one of Europe's leading IT research, analysis and consultancy organisations. We explain how to bring greater Agility to corporate IT systems through the effective governance, management and leverage of Information. We have built a reputation for 'telling the whole story' with independent, intelligent, well-articulated communications content and publications on all aspects of the ICT industry. We believe the objective of telling the whole story is to:

- Describe the technology in context to its business value and the other systems and processes it interacts with.
- Understand how new and innovative technologies fit in with existing ICT investments.
- Look at the whole market and explain all the solutions available and how they can be more effectively evaluated.
- Filter "noise" and make it easier to find the additional information or news that supports both investment and implementation.
- Ensure all our content is available through the most appropriate channel.

Founded in 1989, we have spent over two decades distributing research and analysis to IT user and vendor organisations throughout the world via online subscriptions, tailored research services, events and consultancy projects. We are committed to turning our knowledge into business value for you.

## About the author

---

**David Norfolk**  
Practice Leader  
Focus Area: Development



David Norfolk first became interested in computers and programming quality in the 1970s, working in the Research School of Chemistry at the Australian National University. Here he discovered that computers could deliver misleading answers, even when programmed by very clever people, and was taught to program in FORTRAN. His ongoing interest in all things related to development has culminated in his joining Bloor in 2007 and taking on the development brief.

Development here refers especially to automated systems development. This covers the technology including acronym-driven tools such as: Application Lifecycle Management (ALM), Integrated Development Environments (IDE), Model Driven Architecture (MDA), automated data analysis tools and metadata repositories, requirements modelling tools and so on. It also covers the processes behind them and the people issues associated with implementing them. Of particular interest is organisational maturity as a prerequisite for implementing effective (measured) process and ITIL (v3) as a framework for automated service delivery.

David is a past co-editor (and co-owner) of Application Development Advisor and associate editor for the launch of Register Developer, and is currently executive editor for GEE's "IT Policies and Procedures" product. He has an honours degree in Chemistry and is a Chartered IT Professional, has a somewhat rusty NetWare 5 CNE certification and is a full Member of the British Computer Society (where he is on the committee of the Configuration Management Specialist Group).

His early career involved working in database administration (DBA) and operations research for the Australian Public Service in Canberra. David then returned to his UK birthplace (1982) where he worked for Bank of America and Swiss Bank Corporation, at various times holding positions in DBA, systems development method and standards, internal control, network management, technology risk and even PC support. He was instrumental in introducing a formal systems development process for the Bank of America Global Banking product in Croydon.

In 1992 he started a new career as a professional writer and analyst. Since then he has written for many major computer magazines and various specialist titles around the world. He helped plan, document and photograph the CMMI Made Practical conference at the IoD, London in 2005 and has written many industry white papers and research reports including: IT Governance (for Thorogood), Online Banking (for FT Business Reports), Developing a Network Computing Strategy and Corporate Desktop Services (for Business Intelligence), the Business Implications of Adopting Object Technology (for Elan Publishing).

He has his own company, David Rhys Enterprises Ltd, which he runs from his home in Chippenham, where his spare moments (if any) are spent on photography, sailing and listening to music.

## Copyright & disclaimer

---

This document is copyright © 2009 Bloor Research. No part of this publication may be reproduced by any method whatsoever without the prior consent of Bloor Research.

Due to the nature of this material, numerous hardware and software products have been mentioned by name. In the majority, if not all, of the cases, these product names are claimed as trademarks by the companies that manufacture the products. It is not Bloor Research's intent to claim these names or trademarks as our own. Likewise, company logos, graphics or screen shots have been reproduced with the consent of the owner and are subject to that owner's copyright.

Whilst every care has been taken in the preparation of this document to ensure that the information is correct, the publishers cannot accept responsibility for any errors or omissions.



2nd Floor,  
145-157 St John Street  
LONDON,  
EC1V 4PY, United Kingdom

Tel: +44 (0)20 7043 9750  
Fax: +44 (0)20 7043 9748  
Web: [www.BloorResearch.com](http://www.BloorResearch.com)  
email: [info@BloorResearch.com](mailto:info@BloorResearch.com)