it's about time

White Paper

Kx and Genome Data Mining

Content

1. Executive Summary

The Human Genome Project was the first large scale collaborative venture to uncover the complete sequence of human DNA. Over the span of 10+ years, the challenges and insights gained, has lead to previously unparalleled whole genome sequencing throughput and cost. Currently, in the 'post-genome' era, the major challenge in genetic research is not data collection, rather it is the storage, analysis and transfer of these vast amounts of data. The time taken to analyse this data using standard methodologies can take weeks - reducing these bottlenecks promises to yield significant benefits in wide-ranging areas of biology in areas from genome scans for variants of clinical relevance as well as the use of such information in real-time medical diagnoses.

As cloud computing becomes ubiquitous and concerns around security and privacy are assuaged, collaboration and the use of federated, merged data has become popular. Larger data sets produce more statistically significant results and facilitate the identification of more complex relationships between genome variation and its clinical implications (e.g. disease risk factors). Collaboration in the cloud considerably reduces costs by reducing duplication of hardware resources and IT personnel as well as reducing data transport costs. In recognition of this a number of sequence data repositories have emerged and a number of initiatives are underway. To exploit the exciting possibilities of these initiatives it is clear that bioinformatics personnel will have to work more closely with data scientists.

Kx is the world's most performant database – some performance metrics using commodity hardware are shown in the table below.

Fast Data

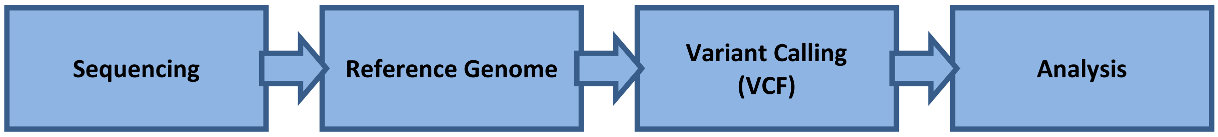
- Single inserts, updates, joins and selects – **millions per second per core. Consistent performance with 10s of billions of inserts per day**
- Bulk inserts, updates, joins and selects: **up to 10s of millions of bulk inserts per second. Trillions per day**
- In-memory table scans of **unrivalled speed measured in milliseconds across trillions of records**
- Supports t**housands of concurrent time series queries** involving billions of rows of data
- Publish/subscribe mechanisms which can update **hundreds of subscribers** or a messaging bus in real-time
- Historical databases allow users to access **terabytes of records** in seconds.
- Nanosecond timestamps

This paper examines how Kx technology provides a compelling alternative to the current mix of technologies for capturing, storing and analyzing data with a focus on genomics, one of the most computationally challenging areas in data science. As well as faster performance kdb+ is a lower cost alternative and provides a unified framework for data collection and storage, data governance, analytics and visualization. Using kdb+ in areas such as real-time analysis of clinical data has the potential to significantly accelerate critical research.

2. Genomics Analysis – Data Challenges

The cost of generating genome data has fallen from billions of dollars to thousands of dollars and an entire genome can be sequenced in days using semiconductors and nanotechnology. Whilst this exciting development has opened up numerous research opportunities in diagnosis and prognosis the volume and structure of the data poses a number of fundamental challenges.

In order to make sense of DNA samples, there are a number of discrete steps which must be executed in a specific order.



Each of these steps has implications from a data perspective;

- The data must be sequenced with the output being small chunks of DNA in digital form. Modern analysis sequences the genetic information using NGS methodologies. This typically produces short (50-200) sequences of overlapping DNA contigs. The relationships between chunks are derived algorithmically to determine their position within the genome. A genome sequence does not solely contain sequential nucleotide identity, but also contains information on the sequencing depth and coverage. This information is important when performing analysis, for both variant positions and non variant positions (those that match the reference genome) and provides indications of data quality. Hence storage of the full genome sequence (gVCF file) is required

- A reference human genome is a representative complete human genome sequence. This genome is typically derived from multiple individuals rather than a single donor and therefore does not accurately represent the set of genes of any single person. A reference genome is used as a common comparator in genetic analysis and research to provide context to variant position, as well as its potential interactions with neighboring variants.

- Variant calling is the subsequent comparison of the sequenced data to a reference and is a key challenge in downstream analysis of this gained genetic information. These tasks are represented as a pipeline which involves many possible bottlenecks including data transfer, analysis and storage. The sequenced genome can then be accurately described in terms of differences to the reference i.e. variant data. This dramatically reduces the storage requirement and subsequent retrieval of the sample data. The resultant Variant call format files (VCF files) are much smaller than the complete sequence assembly – these are typically 350 megabytes as opposed to 1.6 gigabytes.

- The subsequent variant data analysis looks at the distribution of the variant data and often requires summary statistics of a variant occurrence. Hence many queries require an entire scan of the sample data. Analysts then perform a series of informed filters on many fields and annotations to find those few variants of importance. The variant stats are usually analysed in tandem with other meta data such as weight, sex and height to provide additional insight.

There are a number of challenges that a data scientist faces at various stages of this process. These challenges are described below and in subsequent sections we outline how kdb+ is ideal for genomic data analytics and how it is ideal for collaboration at scale.

2.1 Data Analytics

Once the sequenced genomic data has been assembled, current methods for analysing the data use UNIX commands such as grep and awk to parse raw files (gVCF files). Each of the files is approximately 1.6 Gigabytes and available tools have to search the files sequentially, even though many of the queries only require access to specific part of the genome (mapped as chromosome & position). This introduces significant overhead when performing analysis. These searches can take hours. Researchers are generally limited to basic queries because of the lack of a database capable of storing and querying the vast quantity of data in a timely manner.

Furthermore the tools generally used for analytics have not kept pace with the amount of data available. Typical analysis tasks include a range of operations such as slicing the data, filtering, searching, joining with external data sets, aggregating across samples and generating statistical measures. To process 50 genomes and compare results currently can take up to 2 weeks. Many popular tools for analytics such as Excel and SAS clearly do not scale for analysis of large volumes of data. For pipeline processes parallelisation is desirable as is the availability of nested hierarchies – again current tools are clearly inadequate.

Although the time taken to generate sequences has been significantly reduced, the sequenced data is in the form of a series of strings containing hundreds of thousands of nucleotides rather than a single string consisting of billions of orderly nucleotides. It is difficult to represent the complexities and nuances of genome samples without sacrificing analytical capabilities. There is no set standard for the representation of variant information. The Variant Call Format (or VCF) has emerged as a de facto standard but in practice merging VCF files from different sources has proven problematical. This results in a loss of a significant amount

In the next section of this paper we look at how kdb+ is ideally suited for analyzing genomic data and explore the reasons behind this.

2.2 Data Management

The problems of inadequate analysis tools are compounded by the need to collect, store, transfer and manage large data sets. The amount of data generated by sequencing omics will produce storage requirements in the order of exabytes and is rapidly increasing as clinical diagnostics moves into a personalisation era. The cost of storing this amount of data is prohibitive for all but the largest of institutions. It is not only the raw storage cost of the data that needs to be considered. As well as backup storage costs, significant personnel costs can be incurred cleansing data and supporting, maintaining and upgrading systems.

Where cloud/hosted solutions or shared databases are used the transfer of data to local installations can be frustratingly pedestrian. Using standard techniques it is estimated that it takes 20 days to transfer 50 whole genome sequences across the internet, assuming there are no lost connections during the process. Increasing bandwidth connectivity is not an economical option. Bandwidth size and cost as well as contention become an issue.

The heterogeneity of sequenced data generated from various sources is problematical. Differences in sequencing techniques, lack of standardisation and differences in representation are just some of the areas which cause problems. This is exacerbated by the fact that many organisations such as software vendors have a vested interest in maintaining proprietary protocols. Even if standards are adapted they will need to be continuously extended – this has an ancillary impact in the form of constant system upgrades. Integration is required with omics data from other sources such as patient records, billing data and from epidemiological data. The use of standards is an obvious answer but defining common protocols is difficult given the number of player and these protocols take time to become established. The advent of new technology such as the generation of more data by medical sensors and devices will mean that creating standards will be an evolving process which will inevitably lag behind.

Data quality is important as inaccuracy in the original data gets persisted if not cleansed. Duplication and redundancy are also problematical. To be useful the data often needs context and annotation. Metadata needs to be included.

> In a subsequent section of this paper we look at how kdb+ is ideally suited for collaboration at scale between multiple stakeholders. Kdb+ has a low hardware footprint so the Total Cost of Ownership is relatively low compared to other technologies.

2.2 Data Privacy

Data privacy is a major issue as the data can be used to identify individuals and their families. Collection of data without any guarantees as to its future use and repurposing is problematical. There are associated ethical and commercial issues which are exercising regulators and legislators. As the use of federated data sets in a distributed environment becomes more common data quality, provenance and privacy issues become more important.

> We do not explore in detail privacy issues in this paper. However kdb+ has been used to secure highly sensitive data by institutions such as Morgan Stanley and the US Army.

3. Analyzing Genomics Data with Kx

The response time for queries in kdb+ is vastly superior to other technologies. This increases productivity by reducing iteration cycles and facilitates new clinical techniques such as real-time cohort analysis. Kdb+'s speed of query response will enable analysts to attempt new queries previously not possible because of the time they take to run. Our initial findings show that…

Kdb+ is 40 to 400 times faster than traditional VCF analysis

The table below shows the steps taken to create a sample population of 100,000 to test performance of kdb+ and the most commonly used VCF analysis tool in order to test performance for analysis of calculations such as transition/transversion ratio and summaries of indels. A number of other queries were performed to show the flexibility of kdb+ for ad hoc analysis. In all cases the execution time for kdb+ was less than one second.

Traditional VCF analysis	**Kx**
Chromosome Query for one individual • A sample vcf file was downloaded which contained all variants for one sample over the complete genome. The file contained ~51m records. • From this file variant data for Chromosome 1 was extracted (1.8m records), and used as the basis of the tests for the VCFtools records. • Hence VCFtools was not required to scan the entire Genome sample. • The query times displayed below include file opening and closing times. • The queries used were taken from the VCFtools library.	Chromosome Query for one individual • The same single sample vcf file was used to create the kdb+ database. This file contained all variants for that sample over the complete genome. • A kdb+ script was created to read and convert the data. • Each chromosome of the single sample is stored in a separate kdb+ vector.
	Chromosome Query for 100,000 individuals • Further queries were performed on an extended kdb+ database of 100,000 samples. • For these queries the variant data from Chromosome M was extracted and replicated 100k times into one kdb+ database. • Although only a single chromosome was analysed , these queries are an accurate representation of performance, as each chromosome would be stored in its own vector(partition)

3.1 Hardware Environment

The tests were performed on the following hardware running in AWS.

- High Frequency Intel Xeon E5-2670 v2 (Ivy Bridge) Processors I2.2xlarge
- CPU 8 core   61 Gig Memory   2 x 800 Gig SSD

The performance premium for kdb+ is 40-400 times faster depending on the nature of the query.

Query	Execution Time (ms) VCFtools	Execution Time (ms) kdb+
Calculate the transition, transversion ratio of all variants across chromosome 1 for a total of 1,810,839 records.	4131	108
Summarize the indels, composed of all variants across chromosome 1 for a total of 1,810,839 records. Output length of indel, number of occurrences and the percentage of each indel relative to all others.	5036	13

Extrapolating these queries over 100,000 genomes suggests that it would take 2.7 hours Chromosome M.

Extrapolated Query times for 100k genome database	Estimated VCFtools	Estimated kdb+ time
Calculate the transition, transversion ratio of all variants across chromosome 1 for a total of 1,810,839 records.	115 hours	2.7 hours
Summarize the indels, composed of all variants across chromosome 1 for a total of 1,810,839 records. Output length of indel, number of occurrences and the percentage of each indel relative to all others.	140 hours	1300 seconds

Extrapolating these results suggests that it would take 76 hours to analyse all chromosomes.
On a 16 core machine, this time would be reduced to less than 5 hours.

Further queries were performed on an extended kdb+ database of 100,0000 samples. Although only a single chromosome was used, these queries are an accurate representation of performance, as each chromosome would be stored in its own vector (partition). These are typical analysts queries, and the resultant times show the sub-second response of kdb+.

Query	Execution Time (ms)
Pull all records from a database of 4 complete genomes where the nucleotide at position 63,411, chromosome 3 is C	118
Search for all variants on a single chromosome over a database composed of 100,000 samples. Return total.	684
Search for specific variants on a single chromosome over a database composed of 100,000 samples. Return total number per variant.	217
Search for records which pass a quality threshold on a single chromosome over a database composed of 100,000 samples. Return the sample ID, position and sample/reference bases.	241
Search a database composed of 100,000 mitochondrial genomes for the number of each variant per sample.	412

In order to understand why kdb+ produces such impressive results relative to traditional VCF methodologies (and indeed other technologies) it is instructive to examine the properties of kdb+ which make it so proficient at analyzing these types of data sets.

3.2 Why is Kx so much faster for data analytics?

There are a number of reasons that kdb+ is faster than rival technologies for analyzing this type of data.

- Kdb+ has a native 64-bit architecture – essential to manage today's data volumes.
- Kdb+ invokes application logic at the database level. This provides a major performance boost – especially in the case of large datasets – as this removes data transport overhead

Kdb+ is a columnar database

- Kdb+ is a columnar database which makes it much more efficient at retrieving only a subset of relevant data. Variant analysis entails working with chunks of the genome – data quality indicators are required for the first phase of variant calling for example - rather than the full genome so using a transactional database (such as Oracle or SQL) is sub-optimal as the full sequence will be accessed by queries. So, in kdb+, variant data can be represented as a set of features (columns). In many cases just a small subset of features is needed for research (one column)
- Kdb columnar vector storage is ideally suited to storing the genomic data
- As most queries are on a chromosome/position basis, kdb allows the design of a database structure which is specific to the application at hand. For example, a hierarchical partitioning structure.
- The ability of kdb+ to store any size column and any number of columns makes it an ideal solution for the large data volumes required.
- This columnar structure for the database simplifies indexing and joins and dramatically speeds up search performance.
- Hence most mining tasks involve column-specific operations. Column-based storage is inherently read optimized and so facilitates this type of research

Kdb+ is a time series database

- Kdb+ is a time series database and can be used to analyse variations in datasets over time. This will become more important with continued advances in emerging fields such as epigenetics
- As a fully functional relational database it supports foreign keys and ad hoc joins to reference data

Kx is optimized for parallel operations and scalability

- Kx is ideally suited for grid computing and the execution of parallel processes during tasks such as pipeline processing.
- It has built-in multi-core processing and multi-threading. Performance scales linearly with more CPUs, and applications can take advantage of multiple cores without having to write special thread-aware code.
- Kx provides support for parallel access to large partitioned historical databases, so queries can be farmed out to multiple cores/machines.
- Horizontal scalability is linear and virtually without limit. To increase memory or add storage, it is a s simple as adding another CPU or server. There is almost no discernible impact on performance, even with hundreds of CPUs
- Kx's partitioned historical database can scale to tens of terabytes of data with no performance degradation for queries

Kx is superior to Hadoop type implementations because of its small hardware footprint. It does not require banks of super computers with the attendant complexity of managing resources and because the programming needed to analyze the data is easy to learn for people with bioinformatic backgrounds. kdb+ is a unified platform with a simple software stack in contrast to the smorgasbord of open source and vendor supplied software elements.

Kdb+ is used by some of the world's top data scientists

- The technology is used by a thriving community of data scientists with a culture of innovation and mutual support. A wide range of resources including listboxes, forums, public training courses and meetups in locations across the globe are available to new and experienced users
- The technology has been used to build and maintain some of the world's biggest databases and the technology has evolved to meet the challenges of managing and mining these infrastructures.
- Analytical techniques in kdb+ have evolved to meet these demands. A measure of this is the fact that kdb+ has been assessed by the IT industry benchmarking organization, STAC, as being the fastest time series database in 15 out of 17 key measures.

Optimized Functions and Analytics

- Kdb+ is ideal for OLAP queries and drilldown. Many other technologies introduce redundancy and additional maintenance overhead by using database views or precanned queries.
- Kdb+ is ideally suited for clustering algorithms such as cohort analysis and k-means clustering.
- Kdb+ excels at matrix manipulation which is needed for example in hierarchical clustering
- Kdb+ has feature rich graphical interfaces optimized for large data volumes and dynamic OLAP type drilldowns. These graphical interfaces can be extended to visualize data for user defined queries. One popular interface is Excel – queries can be configured in Excel as a client with operations performed on the database and results returned to Excel for display.

Kdb+ provides better performance than commonly used languages **such as R**. These packages load data from files or from a database and there is significant overhead in loading and preparing data as well as persisting any results. For large datasets the difference in accessing data before applying the algorithm and persisting any output could potentially be in the order of days or even weeks.

3.3 A programming language – q

The Kx community is agreed that what sets kdb+ apart from other alternatives for mining large databases is the power, flexibility and ease of use of an embedded programming language. Q has database queries that are similar to counterparts in SQL, as well as functionality that goes beyond traditional SQL

Kx **includes a general-purpose programming language, q,** that has direct support for databases. This gives an enormous advantage over systems that force the user to rely on traditional SQL for data access, or that must depend on a supplier for pre-written queries. With q, the end user can respond quickly to emerging needs:

- It can operate on data directly, minimizing data traffic. There no need to first read data, then export to an external routine for analysis.
- Event processing can be done immediately, as data is received.
- Lists, dictionaries and tables are primitive data types, and the core primitives are designed for this - for example to do arithmetic on tables. An operation can apply just as easily to a million records, as to a single record.
- Date, time, and timestamp (to nanosecond) are basic data types, making time-ordered queries highly optimised
- Data attributes such as sorted can be applied to columns to optimize performance
- the q interactive environment provides immediate feedback for rapid development

The simplicity and elegance of the kdb+ code is evident from the extract below which shows the function for calculating transitions/transversion ratios – a mere 16 lines.

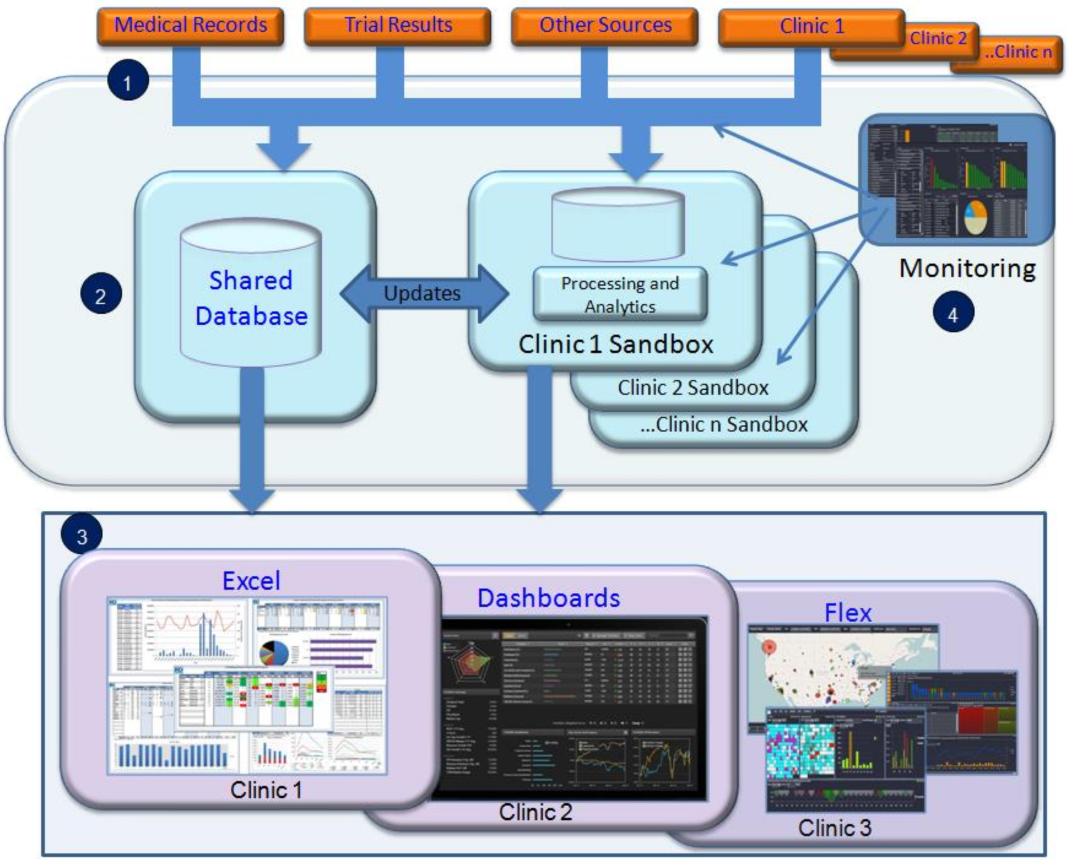
```
kdbTsTv:{[genome]

 //Define substitutions
transitions:("GC";"CG";"GT";"TG";"CA";"AC";"TA";"AT";"AG";"GA";"TC";"CT")!(`tv`tv`tv`tv`tv`tv`tv`tv`ts`ts`ts`ts);
 //All Single Nucleotide Polymorphisms
 snps:select REF,ALT from genome where REF in `A`T`G`C,ALT in `A`C`G`T;
 //All bi-allelic polymorphisms
 multiallelic:select REF,ALT from(update ALT:string ALT from genome) where ALT like "*,*", (count each ALT)=3;
 //Seperate mutations into possible scenarios for bi-allelic polymorphisms
 multimap:delete ALT from(update ALT1:(first each ALT), ALT2:(last each ALT),REF:raze string(REF) from multiallelic);
 allele1:(exec REF from multimap),'(exec ALT1 from multimap);
 allele2:(exec REF from multimap),'(exec ALT2 from multimap);
 tab:(raze string exec REF from snps),'(raze string exec ALT from snps);
 //Sum different subsitutions for;
  /SNPs
   tvsnpsum:(sum(transitions each tab)=`tv);
   tssnpsum:(sum(transitions each tab)=`ts);
  /Bi-alleles
   tvCase1sum:(sum(transitions each allele1)=`tv);
   tsCase1sum:(sum(transitions each allele1)=`ts);
   tvCase2sum:(sum(transitions each allele2)=`tv);
   tsCase2sum:(sum(transitions each allele2)=`ts);
 //Add all SNP + Bi-allelic subsitutions into;
  /transversions
   tvsum:tvsnpsum+tvCase1sum+tvCase2sum;
  /trabsitions
   tssum:tssnpsum+tsCase1sum+tsCase2sum;
}
```

4. Kx for Enterprise Analytics

We have examined how kdb+ delivers significant performance benefits for analysing samples of genomics data. In this section we outline how the enterprise features of kdb+ facilitate collaboration at scale between stakeholders whilst reducing costs and overheads as well as boosting productivity. In short, kdb+ is ideally suited for cloud deployment and Platform/Data As A Service type applications.

Kdb+ can do everything that is required for an initiative of this wide ranging ambition – data capture, cleansing, storage, logging, event processing, analytics and the operation and maintenance of real time and historical databases. A typical architecture for a federated multi-tenanted platform is shown below. Implementation of this architecture helps to address many of the data challenges currently faced by extant systems.



1 Data is ingested either periodically or in real-time by collaborating members and from external sources either...

2 into a shared database or a co-located sandbox (a secure hardware and software infrastructure exclusively for the use of a member).

3 Local applications are used to query and display the results of any operations performed in the data centre.

4 Data, processes, applications and infrastructure are monitored to ensure systems are fully operational and to enable early detection of errors or potential bottlenecks.

Data Collection

As Kx is an open technology the ingestion of feeds from different sources is relatively straightforward. The

- Kx supports loading of files from various formats such as VCF, CSV and TSV.
- Kx is in general open and has a range of well supported APIs including R, Python, JSON, Java and XML. This is also useful when developing applications.
- It is optimized for bulk inserts and updates, and not just one-transaction-at-a time. Genome data typically comes in blocks, and does not have to be written record by record. Bulk inserts of 20m records per second per core are possible while the system is still live without degradation of performance.
- Kx can handle millions of records per second which equates to billions per day and trillions of records in a historical database. The speed copes with spikes in the data flow, and it can also be used with hardware accelerators for maximum speed and flexibility where time is of the essence.
- Kx helps to overcome issues of data homogeneity by mapping on the fly with a suite of embedded transactional and reference data mapping functionality. Cleansing and transformation operations can be performed during this process
- Data can be ingested into both the shared database and private sandboxes simultaneously. A typical architecture for a federated multi-tenanted platform is shown below. Implementation of this architecture helps to address many of the data challenges currently faced by extant systems.

Data Storage

Kx stores data efficiently and works with commodity hardware.

- Kx operates seamlessly with various types of memory including solid state disks and hard drives,
- Kx is ideally suited for a multi-tenanted environment and supports a full permissioning and entitlements framework
- Kx supports lightweight compression techniques such as dictionary encoding and deltas. The benefit of representing data as integers reduces storage costs and cache memory usage.
- Kx supports partitioning of data. This improves performance – the search space for retrieving data from a particular chunk is much reduced.

Efficient storage and compression in Kx has a significant impact. For example, using vcf files as source data in a document type database would increase the footprint

Application Framework

Kx supports an application framework in a multi-tenanted cloud based environment.

- Participants can develop their own applications in isolation from other users while accessing the same underlying database.
- Applications can be developed in a range of technologies due to the open nature of kdb+. Examples include Excel, HTML5, Android, Flex and C#/.Net.
- Kx has interfaces to libraries and statistical packages such as R, Matlab, Python and SAS.
- It is much more efficient to have the applications operate directly on the shared data and accessed remotely through browsers or devices than transporting the data to applications installed locally in the institutions concerned.
- Kx supports large numbers of concurrent users. This is important for collaborative initiatives where there may be large numbers of users accessing the data simultaneously

System Maintenance and Monitoring

Kx has been used for large distributed operations by many of the world’s largest companies, including institutions such as Deutsche Bank and Goldman Sachs. A range of robust, mature enterprise features have been developed to make these infrastructures easy to manage.

Kx Enterprise Features		
Environment Management	**Monitoring**	**Development Tools**
• Realtime sever level signals (CPU, network & Memory usage) • GUI for viewing and managing applications and processes • Database Connection management • Profiling tools to help with hardware, network and software performance • Failover and Disaster Recovery capabilities	• Alerting functionality • Dashboard Visualisation • A specially designed framework for monitoring, backfilling and rebuilding large historical databases • Audit trails • Scheduling and Event Stream Processing and Workflow capabilities to facilitate batch processing	• Analytics and Process Libraries to control and allow reuse of code common to multiple applications • An IDE to facilitate testing and optimization of scripts and queries • Source Code management • Testing management including test data management • Code release management • Debugging
Centralised Configuration Management • Permissions and entitlements at user, group and role-based levels • Configuration of user settings • Enterprise level authentication through LDAP and Kerberos integration		

- Kx is reliable. Many applications written by Kx have been operating 24/7 for years without interruption.
- New analytics can be introduced to Kx on the fly.
- Similarly new data such as additional sequences and reference data can be introduced without impacting system uptime. This high availability is important as downtime for scheduled maintenances and upgrades inhibits productivity.
- By combining the database and application logic layer Kx reduces hardware complexity, fosters productivity and makes code easier to maintain.
- Kx operates seamlessly with applications widely used for unstructured data.

EMEA

Head Office
3 Canal Quay
Newry
Co. Down
N.Ireland
BT35 6BP
Tel: +44 (0)28 3025 2242

Belfast
11-13 Gloucester Street,
Belfast,
Co. Antrim
BT1 4LS
Tel:+44 (0)28 9023 3518

Dublin
Fleming Court,
Flemings Place, Mespil Road,
Dublin 4
D04 N4X9
Ireland
Tel: +353 (0)1 630 7700

London
Cannon Green Building
1 Suffolk Lane
London
EC4R 0AY
Tel:+44 (0)207 3371210

Americas

New York
45 Broadway,
20th Floor,
New York,
NY 10006,
USA
Tel:+1 (212) 447-6700

Toronto
36 King Street East,
4th Floor,
Toronto, ON
M5C 1E5,
Canada
Tel: +1 (647) 256-6626
Tel: +1 (647) 256-6625

Ottawa
300 Terry Fox Drive
Unit 600A
Kanata, Ontario
Canada K2K 0E3
Tel: + 1 (613) 216-9095

Palo Alto
555 Bryant Street
#375 Palo Alto
CA 94301
Tel: +1 650 798 5155

APAC

Singapore
Unit 12-01,
55 Market Street,
Singapore,
048941
Tel:+65 6592 1960

Sydney
Suite 201,
22 Pitt Street,
Sydney,
NSW 2000
Australia
Tel: +61 2 9236 5700

Hong Kong
Two Exchange Square,
8 Connaught Place,
Central,
Hong Kong
Tel:+852 2168 0715

Tokyo
Sanno Park Tower 3F,
2-11-1 Nagata-cho,
Chiyoda-ku,
Tokyo, 100-6162,
Japan
Tel:+81(0) 36205 3494

www.kx.com