



it's about time

Kdb+ FAQ

Contents

Kdb+	4
What is kdb+?	4
What makes kdb+ faster than other big data technologies?	4
Is kdb+ just an in-memory database?	4
What's the advantage of kdb+ over alternatives like open source?	4
Can kdb+ interoperate with other databases and technologies?	5
Which platforms does kdb+ run on?	5
What is the TCO profile of kdb+?	5
Does kdb+ support MapReduce?	5
Does kdb+ support compression?	5
Does kdb+ support multi-threading?	6
Does kdb+ support Image files?	6
What authentication and authorization measures are included?	7
Does kdb+ support unstructured data?	7
Does kdb+ support geospatial data and queries?	7
The q language	8
Is the q query language needed to use kdb+?	8
What are the advantages of q being a vector-oriented language?	8
What kind of resource management does q have?	8
How do you protect against bad queries that may compromise performance for other users?	8
What memory limits are in place for a q session?	9
Can you export data out of kdb+?	9
Connectivity and 3rd Party Interfaces	10
Is kdb+ callable from other programming languages?	10
Can I run SQL queries against kdb+?	10
Do you support WebSockets?	10
Do you integrate into Hadoop and other data feeds?	10
Performance, Scalability & Deployment	11
How do you handle the requirement for high availability?	11
Is there an independent benchmark of kdb+'s speed?	11
Does kdb+ operate on GPUs?	11
Can I run kdb+ in my cloud computing environment?	11
How does inter-process communication work?	12
What Machine Learning Capabilities are provided?	12
Kx Dashboards	13
Can you outline the components of Kx Dashboards?	13
Can dashboards connect directly to the database?	13
How is scalability addressed ?	13

Are the central services transparent from a user standpoint?	13
What sort of admin tasks are required to maintain Dashboards?	13

1. Kdb+

1. What is kdb+?

Kdb+ is the world's fastest time-series database, designed from the outset for high-performance, high-volume data processing. The same in-memory architecture is used for both real-time and historical data and incorporates its own powerful query language, q, so that analytics can be run directly on the data. Kdb+ has built-in multi-threading, multi-processing, and multi-server capabilities.

2. What makes kdb+ faster than other big data technologies?

Kdb+ is a time-series database optimized for big data analytics. The columnar design of kdb+ means it offers greater speed and efficiency than typical relational databases. Its native support for time-series operations vastly improves both the speed and performance of queries, aggregation, and analysis of structured data.

Kdb+ is also faster than other popular databases because it has a built-in proprietary array language, q. This means that it operates directly on the data in the database, removing the need to export data to other applications for analysis. Kdb+ and q make full use of the intrinsic power of modern multi-core hardware architectures.

Finally, kdb+ has an extremely low footprint (<700 Kb) making it highly memory efficient and equally suitable for deployment on edge devices having more limited resources as well as centrally and in the cloud.

The Transitive Comparison paper on the Kx website provides an intuitive comparison of Kx's performance against other technologies [Click here to read the paper](#).

3. Is kdb+ just an in-memory database?

No, kdb+ is also used for storing and analysing large data volumes on disk. Moreover, it is a relational database management system and provides a Complex Event Processing (CEP) engine that operates on streaming data. It is used across multiple industries from storing years of [financial pricing and order data](#) to [billions of meter readings](#) and [telemetry data](#) from thousands of devices.

4. What's the advantage of kdb+ over alternatives like open source?

Functionally, the immediate benefits are the high performance and low latency processing from the combination of in-memory, CEP processing on both streaming and historical data within a single solution. Operationally, the single stack and low footprint of kdb+ ensure lower TCO than open source-based solutions because of ease of deployment and less integration overhead at implementation time, coupled with reduced hardware and easier

on-going maintenance costs

Kdb+, unlike open source alternatives, is supported and backed by a dedicated and responsive development team

5. Can kdb+ interoperate with other databases and technologies?

Kdb+ has open interfaces to enable it to co-exist with other technologies as part of a fabric of big data solutions. It has a number of different APIs for interfacing to Hadoop and other data platforms. Users regularly develop new interfaces and upload them to the Kx Community website so the list of available interfaces is continuously growing. As well as ODBC and JDBC connectors for standard database technologies there are interfaces to C/C++, Java, Python and .Net to help with migrating data between kdb+ and traditional databases or applications like Excel.

Please visit <https://code.kx.com/v2/interfaces/> for more information on interfaces to and from kdb+

6. Which platforms does kdb+ run on?

kdb+ runs on industry-standard 64-bit architectures running Linux, Windows, Solaris (SPARC and Intel), or MacOS.

7. What is the TCO profile of kdb+?

A significant component of the total cost of ownership of a technology stack is the cost of the hardware to run it and the number of people required to install, develop and support solutions build upon it. Our customers tell us that the total cost of kdb+ is lower than that of its competitors because of the small investment in computing resources required to operate it. Comparisons show kdb+ requires 10 to 50 times less computing infrastructure and therefore reduces investment and running costs. It is uniquely engineered to efficiently use commodity hardware and storage devices in any topology – local or distributed – providing flexible and cost-effective ways to add capacity when required.

In addition to the smaller number of personnel required to manage the infrastructure customer, feedback indicates that kdb+ developers have been found to be at least five times more productive than developers working with traditional database technologies. This is due in part to the single stack nature of kdb+ that contrasts to the multiple layered nature of competitors and open source alternatives

8. Does kdb+ support MapReduce?

Kdb+ has always supported MapReduce through the use of its peach (parallel each) functionality. [Click on this link to read more about parallel processing in kdb+.](#)

9. Does kdb+ support compression?

Kdb+ supports compression of both streaming data and data stored on disk. In each case there is a choice of algorithms and a broad range of settings, driving down storage requirements and latency, and optimizing CPU usage.

Note that it is also possible to apply compression on a field-by-field basis, i.e. to use different compression rules from one table or column to another. This can be important for different data profiles where different algorithms may provide greater compression ratios. Kdb+ also supports hardware compression boards (for example, GZIP accelerators) to improve compression/decompression performance.

For more information [please click here to read our white paper on data compression](#)

10. Does kdb+ support multi-threading?

Yes, multi-threading is supported to leverage a multi-CPU processing environment in areas such as MapReduce (LINK).

Please click on [this link to read more about multi-threading](#) on code.kx.com or on the following link to access our white paper "[Multi-threading in kdb+ performance optimisations and use cases](#)"

11. Does kdb+ support Image files?

Yes, image files can, like any other file type, be stored efficiently in kdb+. As stated previously, while kdb+ is optimized for processing time-series data it can perform analytics on aspects of image data too. For example, it is used by one client to process raster bands in multispectral satellite data such as Landsat imagery. Click on this link [Using kdb+ to process Satellite imagery](#) to read more about how it is used.

12. What authentication and authorization measures are included?

Authentication and authorization can be provided by the application or via LDAP or SAML.

If applied within the application, all sensitive data is encrypted in transit and at rest. All stored passwords are hashed with a SHA256 algorithm. A full and configurable authentication policy can be applied at both a user and user group level. This policy meets all recommendations from OWASP

13. Does kdb+ support unstructured data?

With the release of kdb+ 3.6 [Kx provides rapid access to unstructured data](#)

In particular, the new 'anymap' capability allows developers to query unstructured data held in kdb+ much more rapidly. This makes it easier to combine structured and unstructured data within a kdb+ database and analyze them both with the record-breaking speed that kdb+ is known for.

14. Does kdb+ support geospatial data and queries?

Yes [geospatial](#) integration with is enabled in kdb+ through an integration of the [S2 geometry library](#)

The q language

15. Is the q query language needed to use kdb+?

A basic knowledge of q is needed to set up and administer a kdb+ database. Our experience is that most developers find q easy to learn and incredibly productive — even those whose prior experience is solely with traditional programming languages. Since q includes a SQL-like subset, it is very familiar to database programmers, who can be up and running within days.

The query language allows powerful ad hoc queries on-the-fly. This is important as companies are being faced with a huge number of ad hoc reporting and research requirements, putting significant pressure on their data infrastructures.

Also, while end users may access kdb+ from a GUI, and know nothing of the database back end, in practice, many actually prefer to work directly with q in a kdb+ REPL session, making use of the ability to write their own custom queries.

16. What are the advantages of q being a vector-oriented language?

Vector programming languages make calculations and analysis quick and easy by applying operations to a set of values at once. They remove the requirement for loops by enabling calculations to apply to large portions of data in one step. Vector oriented languages are extremely scalable and suited to parallelism due to the simplicity of the operation required and the ability to run in memory. An example would be the ease of calculating the average test score as simply the sum of a vector divided by that count of the vector rather than looking at each score in turn.

Click on these links, [Vector Processing Languages: The Future of Big Data Analytics](#), and [Data Analysis with Vector Functional Programming](#) to read more about the advantages of vector processing languages

17. What kind of resource management does q have?

q offers multiple techniques to manage resources. Process memory utilization can be controlled. Access control and authentication can be applied at either kdb+ level or at enterprise level via LDAP or Kerberos. The Kx platform also offers various ways to manage the number of queries ran through load balancers and gateways.

To know more about the Kx platform [click on this link to the kx Website](#)

18. How do you protect against bad queries that may compromise performance for other users?

We provide the ability to lock down systems to functional API access only. This prevents string queries being executed within the Platform for all users except administrators. In this way we can ensure that only APIs which has been developed and tested for both correct behaviour and performance by the product team can be executed within a system.

19. What memory limits are in place for a q session?

The default workspace limit is 0 (no limit) but provides a number of configuration options

- A `-w` command to limit the amount of memory used by a kdb+ process
- In Control for Kx there is the ability to set the memory limit in the process reserved parameters.
- Options within garbage collect mode using `-g` and passing a 1 (immediate) or 0 (deferred)
 - If immediate is selected memory will be returned to the OS as soon as it is no longer referenced or is hitting an associated overhead. This may offer more control and be easier to manage memory.
 - If deferred is selected memory will return to the OS when the garbage collect is ran or when allocation fails. This may offer better performance.

20. Can you export data out of kdb+?

Data can be exported to a range of formats including CSV, Excel, XML, and JSON. For a list of export types and interfaces please visit <https://code.kx.com/v2/interfaces/>

Connectivity and 3rd Party Interfaces

21. Is kdb+ callable from other programming languages?

Yes, there are interfaces to C/C++, Java, .Net, R, Matlab, Perl, Python and other third party toolkits like Keras, Deepmind, TensorFlow all of which are available as open source, and are supported for our customers.

Kdb+ also supports ODBC and JDBC to help with migrating data between kdb+ and traditional databases or applications like Excel.

[Click on this link](#) to view comprehensive documentation on interfacing with other technologies.

22. Can I run SQL queries against kdb+?

Users can execute SQL queries against kdb+ databases but users predominantly use q, the built in query language that comes with kdb+.

Our experience is that most developers find q easy to learn and incredibly productive — even those whose prior experience is solely with traditional programming languages. Since q includes a SQL-like subset, it is very familiar to database programmers, who can be up and running within days.

23. Do you support WebSockets?

Websockets have been supported natively in kdb+ since v3.0. The function `.z.ws` is called for every client message, which is set by default to echo the message back to the client. For more information on websockets please read our white paper on <https://code.kx.com/q/wp/websockets/>

24. Do you integrate into Hadoop and other data feeds?

Kdb+ has a number of different API's to interface with Hadoop and other data feeds so the user can extend the list of available interfaces. Users regularly develop new interfaces and upload to the Kx Community website so the list of available interfaces is continuously growing.

Performance, Scalability & Deployment

25. How do you handle the requirement for high availability?

Many customers have implemented large, fully-redundant systems. Kdb+ supports failover, so there is no loss of data or performance. There is local logging as well as complete replication between data centers. IT departments can deploy kdb+ in distributed environments, like clouds, and dynamically allocate system resources to meet real-time spikes, such as unusual peaks in market data. There is no need to over-invest in big hardware dedicated to kdb+, while there is extra capacity immediately available when needed.

26. Is there an independent benchmark of kdb+'s speed

It was precisely because banks were dubious of false claims by vendors that the Securities Technology Analysis Centre (STAC) was set up. Its purpose is to provide an independent, audited evaluation of technology performance over a standardised set of benchmark tests. The challenge with performance tests is comparing like with like and filtering out where high performance was arrived at by fine tuning or optimizing in one area to the detriment of another. STAC runs the same test on the same environment to ensure a like for like comparison

For more information on STAC and how Kx has excelled in its performance tests please visit <https://stacresearch.com/>

27. Does kdb+ operate on GPUs?

Kx, via its low footprint and in-memory processing draws optimum performance from CPU's and, to date, has not had to use GPU to improve performance. It does, however, provide an interface that enables q to invoke CUDA code on NVIDIA graphics cards (CUDA is NVIDIA's API for interfacing to some of its GPU instruction sets. As a case study, a Libor Monte-Carlo portfolio computation that took about 26 seconds on a single core of an x86 machine was executed in 0.2 seconds on the graphics card.

It should be noted that because data is copied to the card where the code executed and the results are copied back it should only be considered in cases where the GPU gain outweighs the transmission overhead

EmbedPy enables machine learning libraries to utilize GPU resources from within kdb+.

28. Can I run kdb+ in my cloud computing environment?

Yes kdb+ runs on clusters, grids and clouds including AWS, Azure and Google Cloud.

Kdb+ is available on both [AWS Marketplace](#) and [Google Cloud Launcher](#)

[Click on this link](#) to read more about cloud deployment of kdb+

29. How does inter-process communication work?

Kdb+ has native functionality that allows each process to connect to other processes on the same server or different servers, and allows a time limit to be set on those connection attempts. kdb+ also has native functionality which can be used to protect processes from unfamiliar connections or to password protect any process. These characteristics make grid computing easy to set up, and easy to maintain in a secure manner.

Kdb+ also has some built in utilities for maximising the hardware available to each process. Any process can use 'slave' processes on any server in the grid which are dedicated to providing support to their master processes. These slave processes can also make use of their own slaves to maximise their use of the server upon which they are based.

SSL/TLS are supported for security of data in transit.

30. What Machine Learning Capabilities are provided?

Kx have built a number of tools and features for ML in kdb+ <https://code.kx.com/v2/ml/>

Kdb+ naturally fits very well with machine learning due to its ability to process vast amounts of data using less computing resources than other technologies. Products like embedPy make machine learning projects easier. EmbedPy essentially loads Python into kdb+. This allows access to all of the major libraries that are useful in machine learning such as scikit-learn, TensorFlow and PyTorch.

The implementation of JupyterQ is another useful tool as it provides access to Jupyter notebooks inside q. This is extremely beneficial for visualising data which is vital in machine learning. Both embedPy and JupyterQ can easily be installed alongside kdb+ using a variety of mechanisms, including direct downloads, Docker images, or the Anaconda Python distribution. Kx's ability to process large amounts of data at high speed coupled with the readily available machine learning libraries and visualization tools offers a simple and effective solution for Machine Learning projects.

Kx Dashboards

31. Can you outline the components of Kx Dashboards?

Between the dashboard and the q databases there is a Tomcat server, a query manager and a Control server.

The Tomcat server is the communication layer between the Dashboard front-end and the kdb+ back-end. Managing the flow of information is a query manager; it's role is to optimize performance by caching query results so the kdb+ back-end is not overloaded; for example, by users running simultaneous polling from a shared dataset.

The Control for Kx module acts as an authentication and configuration management framework. Authentication, entitlements, centralised configuration, database connection settings, and dashboard application settings are persisted in Control for Kx. Control for Kx also persists tables and logs to track user-logins and query profiles.

32. Can dashboards connect directly to the database?

A standalone version of Dashboards is available for building simple charts and tables which can connect directly to kdb+ but lacks some of the best features which Kx technology offers. The Kx Platform includes all requirements to maximize the power of Kx including content management, real-time data streaming, and analytics for enhanced security.

33. How is scalability addressed ?

With a single tomcat instance capable of servicing 1,000 concurrent users, it is likely that bottlenecks will appear at the kdb+ end before the UI suffers performance problems. If a client has an enterprise kdb+ license, the back-end cores can be scaled without additional license considerations. The Apache and Control services scale horizontally – additional instances can be added to service more users and more queries as required.

34. Are the central services transparent from a user standpoint?

More specifically Do I need to know anything about it when creating a new database and writing a dashboard to visualize its content? The user does not need to interact directly with the services. They can simply create new database connections by configuring access to a kdb+ host:port, and subsequently write/send queries to that database connection as if they were directly bound to the process.

35. What sort of admin tasks are required to maintain Dashboards?

The query manager is implemented in tomcat. The Control server is a q framework. They do not require maintenance by the user. Most users choose to shutdown and re-start these services once per week, but this is optional. These components are upgraded in concert with Kx Dashboards and customers typically upgrade their version once per year.