kentik

The Kentik Data Engine

A CLUSTERED, MULTI-TENANT APPROACH FOR MANAGING THE WORLD'S LARGEST NETWORKS

When Kentik set out to build a network traffic monitoring solution that would be scalable, high-precision, real-time, and easy to deploy, its founders realized that success depended on making the right architectural choices. To move beyond the limitations of existing systems, they'd have to cast off the shackles of legacy approaches. So they leveraged cutting-edge techniques and technologies from the fields of advanced big data management and SaaS. At the heart of the service they created, Kentik Detect, is a uniquely powerful and scalable data processing core known as the Kentik Data Engine (KDE). This paper reviews the requirements, architecture, and results that have been achieved with the KDE to date.

THE MODERN NETWORK MONITORING CHALLENGE

Traffic volumes and rates flowing across today's networks are growing fast. Much faster, in fact, than the ability of traditional network management tools, commercial or open source, to keep pace. In particular the systems that measure and document what makes up network traffic, using techniques such as flow-record collection and analysis, are too complex and costly to accommodate current needs for scaling and flexibility. Network operations teams are left without affordable options, and as a result they lack the visibility they need to recognize, diagnose, and mitigate performance issues in anything close to a timely manner.

The challenge operators face in finding effective visibility solutions is that most vendors are applying legacy approaches to what is in fact a big data problem. Big data issues are typically defined by industry experts as exhibiting the "three Vs of data." Network traffic metrics involve all three:

- *Volume:* Network monitoring can generate tens to hundreds of gigabytes of measurements per day – even terabytes in large networks, and petabytes when considering systems that must manage multiple networks.
- *Variability:* Traffic and performance monitoring metrics are often a mix of data types, from flow records (i.e. NetFlow, sFlow, IPFIX) to SNMP, log entries, and even PCAP (packet capture) files. A few of these types follow structured rules, but others are entirely unstructured.
- *Velocity:* Particularly when managing a large network environment, traffic metrics will be generated at rates of millions per second.

Most vendors are applying legacy approaches to a big data problem.

Big data challenges are best handled with big data technologies. And for that, there are many choices available today, both commercial and open source in nature. All are intended to help with one or more aspects of the overall challenge of collecting, organizing, managing, and supporting the exploration and analysis of big data. So the problem isn't finding generic big data tools, it's finding big data solutions that are right for network operations. That involves meeting specific requirements, such as real-time data analytics and multi-tenancy, that can't easily be found in or built onto the available solutions.

Another challenge is the way in which network monitoring and management technologies and products are delivered. The commercial network management tools sector has long been dominated by solutions using proprietary data architectures and expensive, inflexible, appliance-based approaches. To the extent that those solutions can achieve impressive scale, it comes at a wholly unreasonable cost in terms of both capital outlay and ongoing support and maintenance. The broader management tools market has embraced SaaS as a delivery model for advanced products and capabilities, and it's time to apply the same strategy to network traffic visibility.

The Kentik team defined essential requirements for the operations use case.

USAGE REQUIREMENTS

Fortunately, the team came from backgrounds in direct operations and had experienced first hand the lack of effective, scalable flow monitoring tools for operations. Discussions with their peers confirmed that many others were dealing with the same problems. Some were trying to build their own systems, using a mix of open source and commercial tools, while others were simply doing without. Determined to close this gap, the Kentik team started with a number of essential requirements for addressing the operations use case:

- *Ingest scalability:* Inbound capture would need to accommodate extremely high volumes of data, on the order of billions to trillions of data points per day and multiple millions of data points per second. Ingest at that rate would require a highly optimized and distributed approach.
- *Retention scalability:* Retaining all raw incoming data for months at a time to provide clear, complete, and accurate "lossless look back" for troubleshooting and forensics was a must.
- *Reliability:* High availability (HA) and high reliability, with internal balancing and rate-limiting would be needed so that no one source of data or investigative query could adversely impact others.
- *Real-time:* Support for continuous review and sub-minute alerting against inbound data was essential, allowing network and security operations to be informed of changed conditions in time-frames matching current business and consumer expectations: immediately, not ten or twenty minutes after the fact.
- *Easy to use:* A fast intuitive interface that minimizes impediments to troubleshooting and diagnostics in high stress situations such as an outage or major degradation was an absolute necessity.
- *Responsive:* Rapid response to data exploration queries across huge volumes of data is another 'must-have.' Capturing data is one thing, but getting that data out in a timely manner, and then being able to quickly filter and explore very large data sets is also extremely important.
- *Simple deployment*: Ease of deployment and configuration, without extensive customization services needed to be a guiding principle. The alternative is systems involving lengthy, expensive installation and needing costly ongoing maintenance, which often results in falling behind on patches and upgrades due to the required effort and the risk of breakage.
- *Multi-tenant:* Long required in the provider world, multi-tenancy is increasing needed by enterprises as well, so that data and activity views can be cleanly separated within the system to prevent leakage and impose access controls.
- *Open access:* Data held within the solution must be accessible not only for real-time monitoring and alerting but for sharing with other existing internal and third party systems. Access should be secured, but must also be quick and easy.

A solution that could meet all of these requirements would be ideal for supporting both the real-time operational and long-term engineering/planning needs of organizations with very large networks. And if done right, it could also be made to fit easily and cost-effectively into the scope of needs for any size operations team.

ARCHITECTURAL REQUIREMENTS

The only effective approach is a distributed, clustered solution.

The Kentik team realized that meeting the requirements outlined above would be no small order. It was quickly determined that the only effective approach would be a distributed/clustered solution. As the usage requirements above were further refined, the following specific architectural requirements emerged:

- Support for constant high speed ingest, to trillions of records/day per cluster.
- Back end storage and data management at multi-petabyte scale.
- Replication on ingest and resync on disk/node failure, for durability and HA.
- Data indexing for fast sub-selection without linear scan.
- Compression support for efficient storage (and I/O read speed) of data.
- Subquery caching/memorization for reuse without re-querying.
- Subquery rate-limiting to protect the cluster from large unbounded forensic queries.
- Support for in-memory storage of recent data to enable continual low-latency real-time anomaly scanning.

In parallel with the above, the team also believed strongly that now was the time to bring the SaaS model to the network monitoring and management sector. A SaaS approach would drastically simplify deployment and administration. And by providing the back end infrastructure as a service, SaaS could avoid scaling issues much more readily than on-premises deployments. While the SaaS approach had its clear advantages, it significantly elevated the requirements around data security and multi-tenancy.

ALTERNATIVES CONSIDERED

The team was unable to find an open source or commercial flow system that met the requirements.

The team searched far and wide, but was unable to find any open source flow system with the required clustering support and multi-tenancy features, particularly when considering subquery caching and rate-limiting. Candidates considered included Hadoop and other open source big data solutions, all of which exhibited disqualifying issues with ingest rate and/or query times. Similarly, the team could not find a commercial enterprise solution (e.g. Teradata, Vertica, etc.) with an acceptable cost/capacity relationship, given that deployments would quickly reach multiple petabytes.

Conceptually, the team was looking for something architected like Google's Dremel. But because NetFlow-like data is "wide" compared to most types of Internet of Things and SNMP-like data, Impala and Drill were not able to support the required levels of ingest, capacity, and multi-tenancy. The remaining options were no more promising:

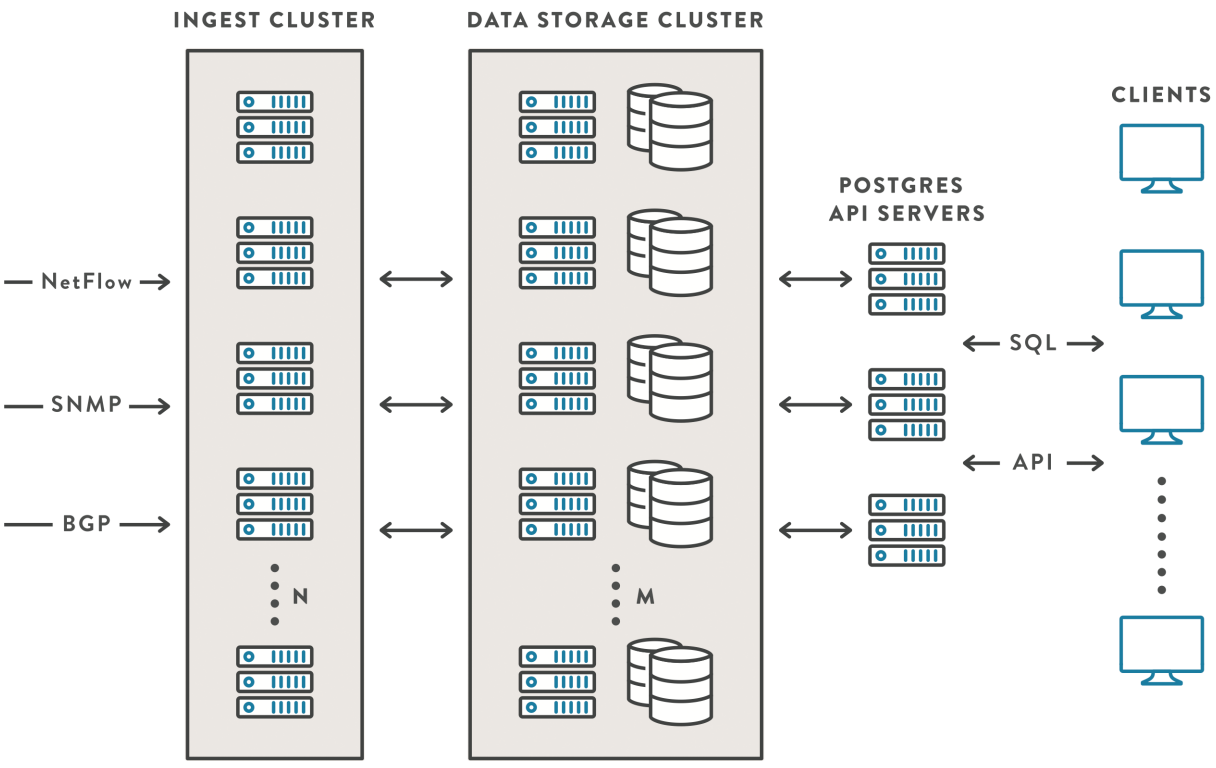
- Older Hadoop-ecosystem SQL solutions like Pig and Hive were too slow.
- Key:value stores don't capture richness, which is requiring pre-build aggregates.
- Simpler time-series databases aren't designed to do massive queries and group-by operations on tags.

SOLUTION: THE KENTIK DATA ENGINE

Given the lack of viable open or commercial components, the Kentik team proceeded to design and build the Kentik Data Engine (see figure 1), a back end that meets the challenges and requirements for both architecture and usage. Key elements of the KDE architecture include:

- A clustered ingest layer currently receives flow, SNMP, and BGP data (additional data types to follow). The data is unified and indexed into a second cluster of data storage nodes. Data is stored in columnar format sliced by time for months to years, and also to a set of in-RAM machines for more recent data.

Fig 1: Kentik Data Engine architecture.

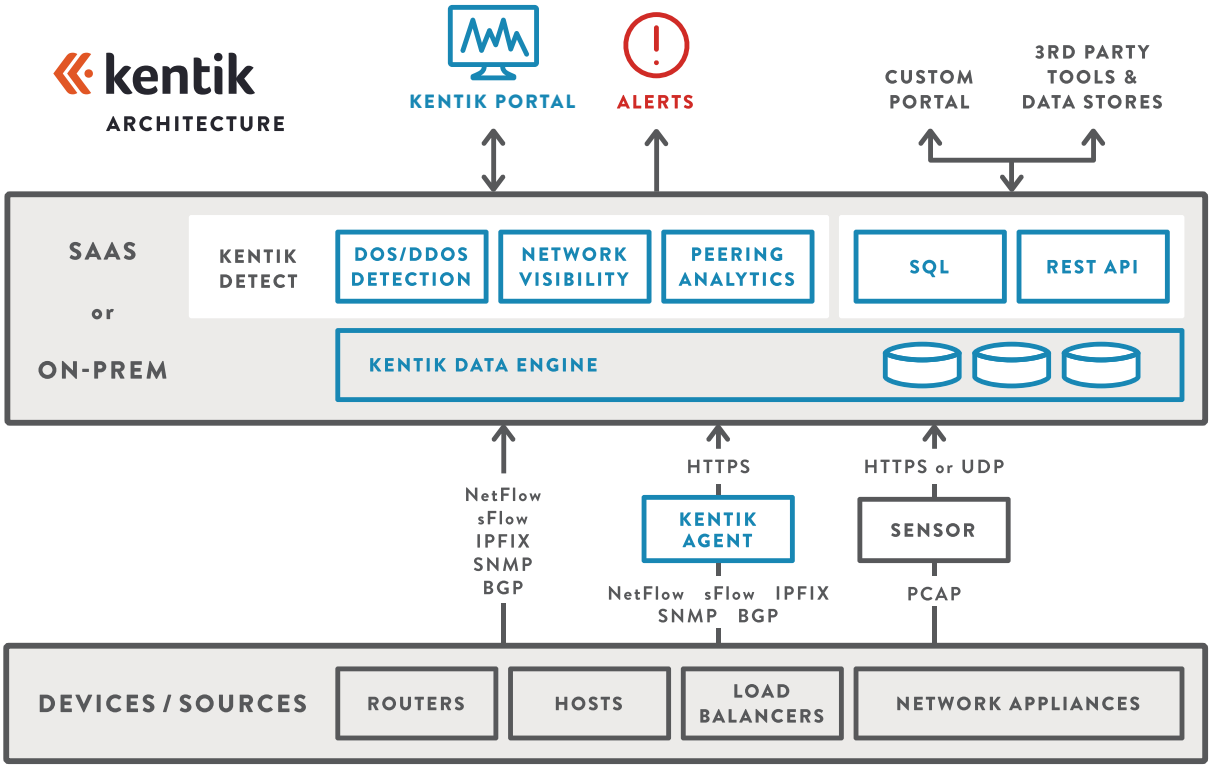


- A front end/API in PostgreSQL. Queries are parsed by Postgres and passed to the KDE system, which breaks up each query by time and target, checks a persistent subquery cache, doles out subqueries to the backend nodes that have the data, and combines the results.
- A scale-out metadata layer, used to track data across the nodes from index time to optional purge.
- Caching of query results by one-minute and one-hour time periods to support sub-second response to operational queries and alerts across tens of Terabytes.
- Full support of compression for file storage to provide both storage and I/O read efficiency.
- Rate-limiting of ad-hoc, un-cached queries to guard system usability against large new queries that require scanning massive on-disk data sets for the first time.

KENTIK DATA ENGINE IN PRACTICE

The Kentik Data Engine sits at the heart of the Kentik Detect service, which delivers real-time network visibility, DDoS detection, and peering analytics (see Fig. 2). KDE handles large-scale data collection via a variety of data paths inbound, and supports access and visibility via the Kentik portal, automated alerting, and SQL or REST APIs.

Fig 2: Kentik Detect system architecture.



The KDE back end went into production for Kentik Detect in June 2014 running on a cluster of machines in Ashburn, VA. Additional KDE instances have been established for Kentik Detect in San Jose and Amsterdam, and more are planned. As of June 2015, the system is concurrently supporting dozens of user networks with hundreds of devices, creating over 40 billion flow records/day, peaks of millions of flows per second of ingest, and trillions of accessible records on the production cluster. The portal front end retrieves all data by executing SQL queries, and can continually load pages across all concurrent requests in a few seconds or less.

KDE was also designed for portability. Several on-premises deployments have been successfully completed on privately owned networks for use by network operators and large scale enterprises. The KDE "stack" is the same on those deployments as on Kentik's public Kentik Detect cloud infrastructure.

CONCLUSION

KDE leapfrogs the barriers that have long limited legacy approaches.

Effective real-time network visibility is a never-ending and ever-growing challenge that has outpaced the rate of innovation by traditional network management vendors and products. With Kentik Detect, Kentik is taking a new approach, applying fast-maturing big data and SaaS technologies to benefit network and security operations. The essential core of the Kentik solution is the Kentik Data Engine, a clustered big data platform that leapfrogs the scalability, flexibility, and cost-effectiveness barriers that have long limited legacy approaches. The Kentik Data Engine will also serve as an ideal platform for future growth and expansion of Kentik's service and solution offerings.

To learn more about Kentik Detect and how it can address your network visibility needs, please contact sales@kentik.com.

KWP2-150721