DEVO

Technical Platform Overview

WHITE PAPER

The Devo Data Analytics Platform unlocks the full value of machine data for the world's most instrumented enterprises, putting more data to work—now. The platform makes extraordinary performance claims relative to competing solutions; in fact, the Devo Platform is 20 to 50 times faster than existing machine-data solutions. For every 64-core data node, the scalable Devo platform:

- Ingests a total of **2 TB of data per day**, collecting up to **1.2 million events per second**
- Accesses **48 million events per second** in queries
- Services **4,000 concurrent queries**

Using a no-compromises architecture designed to scan more data faster, and to scale as data grows using a fraction of the hardware required by other solutions, Devo avoids the bottleneck of indexing data at ingestion without paying the penalty of slower query response. The Devo architecture is built for scalability and performance.

This paper describes the Devo platform architecture, how it is purpose-built for growing volumes of machine data, and how Devo achieves performance numbers that differentiate it in the marketplace.

EXECUTIVE SUMMARY

Devo takes advantage of modern computing infrastructures that leverage high network bandwidth and commodity hardware to enable massive parallelization. Architected as a cloud-first, horizontally scalable platform, Devo is optimized to run on commodity servers that offer fast I/O and reduced access latency. The system also takes advantage of all available storage tiers. The freshest data is available in system memory, then the system uses local flash storage—such as NVMe—and then data is stored on connected storage, such as block storage or external storage devices. All data stored on any tier is hot, for the most available and fastest access.

MASSIVELY PARALLEL

Devo was designed to take advantage of parallelization. Everything in the platform—ingestion, storage, data classification, tokenization, compression, and all query access—is optimized to use parallelization.

HORIZONTAL FIRST

Optimizing the platform for commodity servers rather than proprietary systems, and architecting for parallelization and horizontal scale, results in a cloud-native solution with no limits on the platform's ability to run in any environment.

THE PROBLEM DEVO SOLVES

Devo improves access to data across the instrumented enterprise, both streaming and historical, in real time and at scale, enabling businesses to use this data to reveal the analytic insights contained in machine data from all sources—devices, sensors, infrastructure, applications, and users. The platform is built for scalability and performance, in particular for ingest and query operations. The architecture is optimized for search, with a focus on being able to conduct many simultaneous searches. How Devo ingests data, and what that means for data access, is described next.

HOW DOES DEVO DO IT?

The foundational components of Devo allow us to deliver a data analytics platform with the speed, simplicity, and scale required by the modern enterprise. The Devo technology stack delivers this breakthrough performance while also reducing operational costs significantly.

Devo was designed to be index independent. The platform does not index data at ingestion, which increases scalability and performance. Competing solutions index data on ingest and rely on large indexes, which are more expensive to update and cause ingest rates to decrease as index size grows. Maintaining one big index is slow—at ingest, adding data makes it slow, and when searching, it's more computationally expensive to search. Big indexes also cause performance impacts in other parts of the application, for example via slower response times, which ripple down to slower user access to data. Applications built on competing solutions that need access to historical data to make sense of streaming data will see material impacts on performance.

In addition, solutions built to index data on ingest require dedicated CPU and memory resources that cannot be made available for query. As a result, for high ingest rates, these nodes can become CPU-bound, resulting in diminished performance and requiring additional nodes to support search/query functions.

The Devo approach, in contrast, reduces hardware costs and administrative burden by creating many micro-indexes in parallel, delivering high performance and predictable, fast response rates to real-time queries on large data sets. These micro-indexes, built asynchronously after data is ingested, take advantage of parallelization to eliminate the performance bottlenecks common to solutions architected with a single, large index. For example, to manage 7 TB of data, with 1,000 concurrent queries, 500 concurrent users, and data retention of one year (hot data), Devo requires 9 servers with 288 cores, and 510 TB of storage. Competing platforms require roughly 75 percent more resources—43 servers, 1,376 cores, and 2,408 TB of storage.

Time is the first parameter used in all data searches across a customer's environment. The time-based classification approach allows the Devo platform to dictate where in the file system classified data will be stored. The next set of parameters for data classification is tenant, followed by tags—typically technology type, then vendor, then by many customizable tag levels.

Data is not transformed or structured, but is immediately written to disk in its raw format, then compressed by 90 percent. This enables Devo to provide several customer benefits: queries use less CPU time; keeping data in its raw form makes it possible to quickly reprocess data without re-indexing for new queries; and performing unions against historical data when formats change is simple.

After data is ingested, an out-of-band tokenized index is created from the raw data and written to disk asynchronously. These micro-indexes sit side by side with the original raw data at the top-most tag level. Each day, throughout the day, the micro-indexes are created and updated. At the completion of the day the micro-indexes are made immutable: build the index once and it is not necessary to touch (add data to) the index again.

Having one micro-index created per day per source data type enables parallelization of index lookups, speeding and simplifying search. The query engine can be precise about the number of micro-indexes it needs to read or search through. Limiting index size ensures there is less contention for writing an index while a user is also trying to read it. Micro-indexes can also be aggregated as needed - queries can combine and use micro-indexes from technology types to return larger datasets per query.

Organizing data by several dimensions and tokenizing it to make it easier to search and query enables Devo to stitch micro-indexes together at query, enabling linear performance and scale.

DEVO DOES NOT INDEX DATA AT INGESTION. IT STORES DATA IN ITS RAW FORMAT, ORGANIZED FIRST BY TIME—YEAR, MONTH, DAYS—AS WELL AS BY CUSTOMER DOMAIN, SOURCE, AND TYPE.

COMPRESSION

The platform's approach to compression is optimized for modern architectures with fast CPUs. All raw data is compressed, typically at a 10:1 ratio, using the LZ/LZW algorithm. Experience has demonstrated average compression rates of 90 percent or better. High compression enables query processes to read less data, maximizing search performance even with relatively slow disk storage. Many queries are CPU-intensive and done in-memory. Thus, by frequently compressing data, the system reads less data off disk, which is slow, and does more processing in the CPU, which is fast. Reads of select datasets only are necessary to respond to queries, as described in the next section.

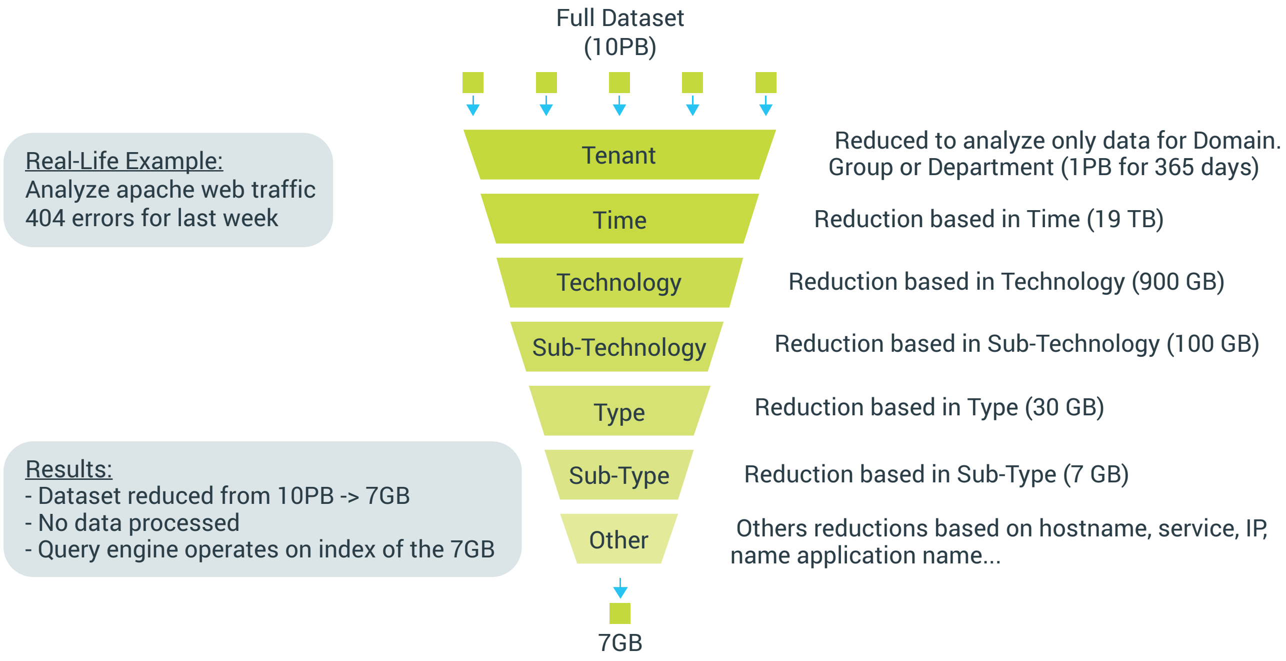
The query engine is optimized for time-based queries to provide predictable, fast responses to queries on real-time and historical data. Parsing occurs only at query time, providing the ability to adapt instantly to changes in underlying data or changes in questions people want to ask of their data, without the need to rebuild indexes or reformat data.

Query processing takes place on data nodes in the distributed architecture. Querying small data sets in parallel using time, filters, grouping, and aggregation quickly narrows down the data that must be read and parsed. Results from data nodes are aggregated into a single data set on meta nodes. Response time is improved, with query latency measured in sub-seconds.

Each data source has a stored parser (or parsers) to decode its data format. To launch a query on streaming and/or historical data, the platform begins by selecting multiple data sets with a time range; data sources (e.g., firewall, log, application); and associated parsers. Data is presented to users in a virtualized table based on data source. Unions of tables enable users to analyze multiple data sets or to enhance the data against lookup tables without the significant overhead caused by joins. Queries can select data from multiple data sources via unions of tables, or multiple table unions. Union tables are a purely virtual construct created at query time—no data is duplicated or moved—meaning Devo looks at the minimum data set for each query. Queries can further reduce dataset sizes by filtering, grouping, or applying aggregate functions.

QUERY ENGINES AND CONTINUOUS QUERY

Data Reduction Optimizes Query Performance



Real-Life Example:
Analyze apache web traffic
404 errors for last week

Results:
- Dataset reduced from 10PB -> 7GB
- No data processed
- Query engine operates on index of the 7GB

Reduced to analyze only data for Domain. Group or Department (1PB for 365 days)

Reduction based in Time (19 TB)

Reduction based in Technology (900 GB)

Reduction based in Sub-Technology (100 GB)

Reduction based in Type (30 GB)

Reduction based in Sub-Type (7 GB)

Others reductions based on hostname, service, IP, name application name...

Query is completely independent of ingest within the data node. From the user's point of view, data is available for query as soon as it reaches memory, only milliseconds after ingest. This approach enables a single 64-core data node to query up to 48 million events per second. The entire query system scales horizontally by adding data nodes.

CONTINUOUS QUERIES

As data is ingested, queries can register to be notified when new data enters the system. This 'push' model takes load off the system because events are fed into the stream, reducing the need for computationally-expensive repetitive querying or polling.

DATA REDUCTION

Devo's approach to ingestion, data classification, and tokenization enables the platform to reduce the amount of data that must be scanned to respond to queries. At ingest, classified data is written to disk and tokenized in a parallel process. Classification enables data to be tokenized and reduced by time, customer domain, data source (technology type and subtype), etc. Data reduction allows the query engine to limit the amount of data it needs to look at to respond to a query. Instead of looking at raw data, the query engine scans the index, extracts targeted data, and then does a full scan of the newly-reduced dataset.

The combination of data ingestion, reduction, and storage methods maximizes search efficiency across the entire platform, regardless of the volume of data at rest.

Data operations platforms that require use of proprietary search processing languages or highly specialized expertise in SQL or JSON effectively limit much of the system's value to expert users only, and make ad-hoc interrogation of data impossible for the average operator. The Devo platform was designed to be intuitive and visual to make access to data available to a larger audience of users.

Devo employs a visually driven data interaction model through which nontechnical users can search, select, visualize, and analyze their own data without writing a single line of code. No knowledge of specialized query languages is required. A simple Navigation panel in the user interface is shown below. The Navigation panel enables users to view which data has been collected, what alerts have been received, and any real-time stats. More experienced users can run queries using LinQ or SQL directly in the UI; queries also can be run via the API, using the exact same query. Devo presents data to users via an abstraction, virtualized tables. At query time, the platform takes information refined by the query process, combines it with the parsers for the data sources, and pushes the information into a visual, easily digested format. A virtual data table is created from the resulting parsed data.

The platform includes Devo Activeboards, data visualization tools that enable users to create, customize, and share data and insights. Activeboards make it easy for users to create visualizations of search queries, manage virtualized tables, and build interactive reports to visually depict the current status of metrics and key performance indicators of data. Simple visual search lets users view, search, modify, and enrich collected data. An Alerts panel gives access to all information on triggered alerts. Notifications are easy to set in the Preferences panel.

Devo also provides multiple service components to make it easier to use the platform. Services include a Correlation Engine; an Aggregation Engine; a Machine Learning Engine; a Query Engine; Alerting functionality; Data Enhancement (lookups) functionality; APIs, and a web-user interface. Services sit on top of the core Query Engine and data model of the platform.

The Devo Platform is a full-stack, multitenant, distributed data, and analytics platform that scales to hundreds-of-petabyte data volumes while broadening the audience that can use data analytics beyond IT and data science.

The building blocks of the platform are simple. Data from every part of the business streams into the data store and is immediately ready for real-time and historical analysis for a variety of use cases. Operations, IT, security, data scientists and other teams can visually interact with and analyze data via a user interface that makes analytics intuitive and fast, even for non-technical users. An industry-standard query language can also be used within the UI itself or via an API, enabling the automation of and integration with other business and operational processes.

The platform offers data isolation, governance, and security while providing the cost and operational savings of shared infrastructure. Data and processing are distributed across a multi-tier architecture consisting of data nodes, meta nodes, and event load balancers. For greater detail on these components of the platform, view the white paper *Speed, Simplicity, Scale*.

The platform uses extensive parallelization—every operation involved with data is parallelized, including ingest, indexing, querying, and compression. This parallelization enables the platform to provide predictable linear scalability using a fraction of the hardware resources needed by existing solutions.

CONCLUSION

Devo designed the Devo Data Analytics Platform to help IT executives realize the transformational power of machine data, with game-changing economics in a no-compromises architecture. Devo creates new business value from untapped data while significantly reducing operational costs.

Learn more at devo.com