CHAOSSEARCH

PLATFORM TECHNOLOGY DEEP DIVE

Store everything. Ask anything.

Thomas Hazel
Founder/CTO, **CHAOS**SEARCH

This paper discusses the advantages and benefits of the **Data Edge** technology and associated architecture backing the **CHAOS**SEARCH platform.

The "**problem**" this data platform has been designed to solve is three-fold:

01 The ability to simply and inexpensively store data at scale.

02 The removal of complicated scaffolding and external systems for data management and analytics.

03 The unification of these aspects into one solution at a disruptive price.

To overcome these challenges, **CHAOS**SEARCH has built, from the ground up, a new distributed database technology and architecture.

At the core of the problem is the complexity and cost of integrating storage with indexing for analytics at scale. Core to the **CHAOS**SEARCH value proposition is the ability to dynamically and seamlessly combine storage with indexing for live and historical analysis, where Data Edge is the new indexing technology with its purposely built distributed **Data Fabric** as the elastic glue. The fabric connects, manages, and orchestrates the Data Edge indexing, search, and query analytic execution as a service.

OVERVIEW

Data Edge

Data Edge (patent pending) is a universal representation for both text search and relational analytics. This format allows for compression ratios below 'theoretical minimums' with built-in 'schema detection, normalization and virtualization' functionality. This format also can be seen as a database index where text search and relational queries are resolved via lazy/late materialization. Unlike traditional database indexing technology, Data Edge constructs are not monolithic and eliminates sharding.

OVERVIEW

Data Fabric

Data Fabric (patent pending) is a cloud agnostic distributed framework for application portability and connectivity. It streamlines, automates, and orchestrates application deployments and resource allocation and management. The fabric can be seen as an elastic administrator for distributed Data Edge indexing, search, and query execution. Each aspect of this elasticity allows for bursting, stealing, and throttling of resources over secure multi-tenancy containers and network overlays.

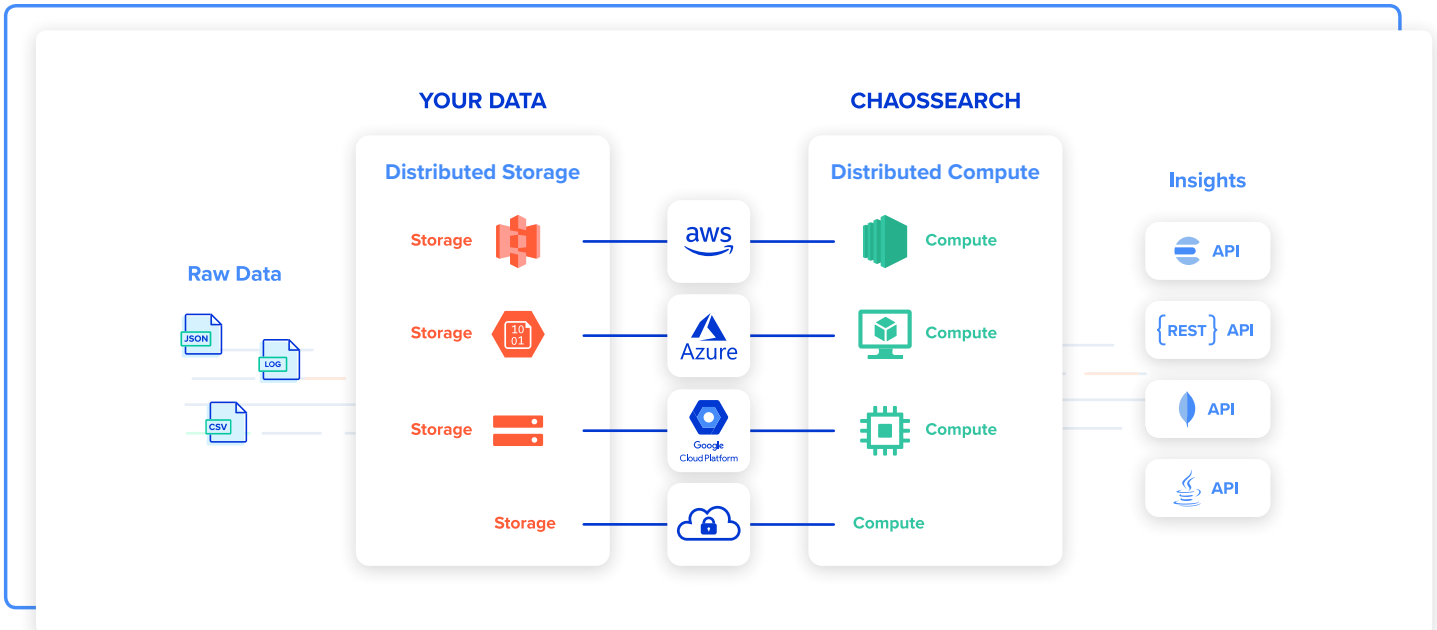
CHAOSSEARCH

Platform

CHAOSSEARCH is a search and data analytics platform, offering both data management and analytics services that enables customers to store more, ask more, and improve analytics and output — at a greatly reduced price.

The platform and associated service is designed to deliver simple, scalable, and inexpensive analytic access to cloud storage as well as information about the storage. This includes what, when, and how much data was stored. All of which enables you to search, query, and visualize data. The **CHAOS**SEARCH difference is simple: **Remove the moving parts and challenges of traditional text search and relational systems and make management easier.** The goal is to streamline the process and dramatically reduce the cost to store, index, search, and/or query.

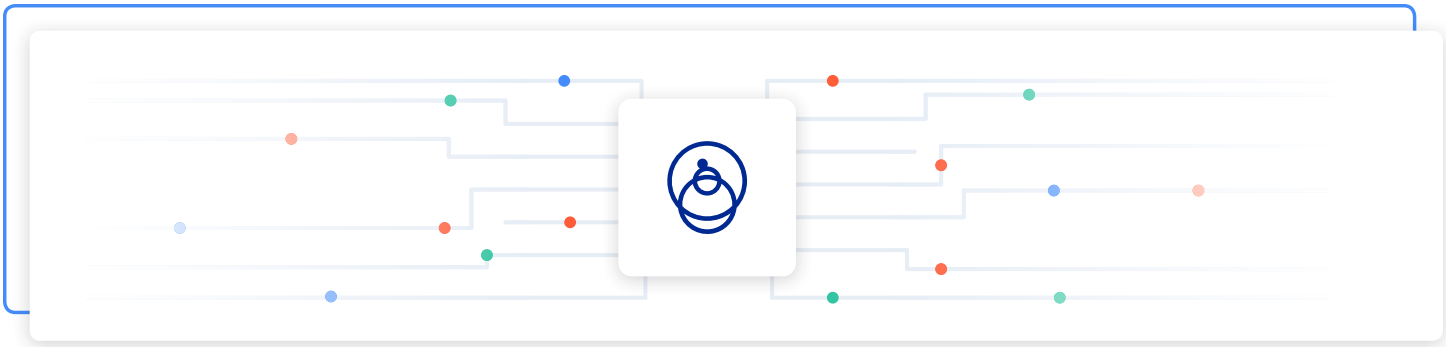
To achieve this, we selected a cloud storage platform that is simple, scalable, and extremely cost-effective, Amazon Web Service (AWS) S3, where the **CHAOS**SEARCH technology and architecture is cloud agnostic. Next, we reinvented indexing and delivered a powerful alternative to traditional Lucene and Columnar approaches. We created Data Edge (and its associated fabric), a new distributed database that discovers, normalizes, indexes, and compresses data without human intervention. Data Edge uniquely supports both text search and relational queries, while, at the same time, optimized for cloud storage accessed over a serverless (separation of storage from compute) computational fabric. The initial focus and use case of the platform is on the analysis of live and historical log and event data. A fully managed service, the platform allows organizations to easily store, search, query, and visualize terabytes of data within their "own" S3 infrastructure. **CHAOS**SEARCH decouples storage from compute, delivering an extremely cost-effective and scalable solution for analytics. The platform automates the cataloging and indexing of data. With both S3 and Elasticsearch API extensions, the **CHAOS**SEARCH platform facilitates data hunting and trend analysis over large datasets by uniquely combining both text and relational analytics through a single console with an integrated Kibana interface.



CHAOSSEARCH

Technology

The **CHAOS**SEARCH platform and service is comprised of two key components: Data Edge indexing technology and Data Fabric architectural framework. Each work hand-in-hand, where Data Edge enables new thinking and possibilities within the distributed computational fabric.



Data Edge

Data Edge was created to address several key challenges typically associated with large scale data analytics. Any one of these challenges can be solved in isolation. However, the ability to address the issues simultaneously by leveraging one core technology was the driving principle. Specifically, Data Edge delivers:

- Index “all” data and “any” type of source
- Built-in auto normalization and transformation
- Built-in compression with ability to be lossless
- Built-in metadata and stats for optimal query planning
- Built-in metadata and stats for optimal distributed execution
- Parallelizable execution model based on share nothing design
- Representation that is amenable to Tensor Machine Learning
- Support both text search, relational queries, and aggregations

Data Fabric

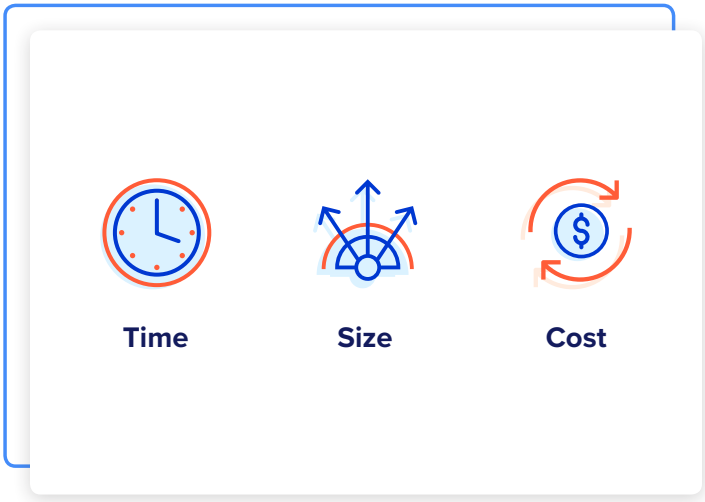
The Data Fabric is based on a Scala/Akka distributed architecture oriented around the strengths of Data Edge capabilities. The Data Fabric is an elastic message-based distributed actor model for orchestrating Data Edge indexing, search, and queries. Key components include:

- Cost analysis metrics for distributed scheduling
- Elastic execution based on workload scheduling
- Built-in intelligent query planning and optimization
- Built-in back pressure throttling from API to/from S3
- Module front end to support multiple search/query APIs

CHAOSSEARCH

Metrics

There are three key metrics that drive the benefits of Data Edge and its associated Data Fabric: time, size, and cost. For instance, how long does it take to index a data source, what is the size of the resulting index, and what is the associated cost? Time, size, and cost will also be outlined when it comes to writing and reading indexed data to and from S3. In this outline, single thread execution will be used to describe the overall metric calculations and ultimately describe the benefits. However, it should be duly noted that the **CHAOS**SEARCH service supports parallel and distributed execution where any node in the fabric can be sliced up or scaled out for cost-effective scheduling and performance.



Data Edge Compression

A key component of the **CHAOS**SEARCH platform is the ability for Data Edge to reduce the size of information while still fully indexing it. The ability to compress raw data sources beyond what has previously been achieved means there are significantly less resources required to back it — everything from storage, network, and compute. The Data Edge representation uniquely optimizes existing compression algorithms. For instance, Data Edge can utilize the SNAPPY compression algorithm, which is fast, but not good at reducing size and make it 2 to 3x smaller. In other words, all the benefits of speed while achieving a size equivalent to the GZIP algorithm where GZIP has good compression but is extremely CPU intensive.

Data Edge Calibration

To understand the benefits of the technology empowering the **CHAOS**SEARCH platform, let's calibrate time, size, and cost metrics based on a specific but classic log use case. Amazon Elastic Load Balancing (ELB) logs will be used as the basis for following metric analysis, where a conceptual 1 TB dataset, which is roughly 3.5 billion entries, will be outlined and calculated. But let's first baseline Data Edge index metrics by performing a single threaded (i.e. one client) solution to solution comparison. As part of this comparison, relative compute and storage are used. However, since **CHAOS**SEARCH is the only solution that supports both text search and relational queries, only "indexing" time, size, and cost will be compared.

These results are based on similar costs, showing some impressive time and size metrics when compared to both Elasticsearch and Redshift. Each use-case will result in different metrics, but this ELB calibration will help put the following 1 TB analysis into context.

Note: The **CHAOS**SEARCH platform uses no local or network storage such as HDDs or SSDs. All storage is 100% on S3 for both writing and reading Data Edge indices.

ELB (35 million logs)	CHAOSSEARCH	Elasticsearch	Redshift
Index - Time	6 minutes	60 minutes	5 minutes
Index - Size	1.5 GB	21.5 GB	3.2 GB
Index - Cost	c4.2xlarge	c4.2xlarge	dc2.large

Data Edge Indexing

Unlike the above calibration and comparison, **CHAOS**SEARCH platform slices up datasets by available compute capacity on the Data Fabric. In this example, the Data Edge "worker" used 1 GB of RAM and 2 CPUs. Therefore, a c4.2xlarge fabric node could be sliced into roughly 4 index workers for a particular dataset and significantly reduce the overall time and cost.

Data Edge indexing, either in parallel or distributed, produces an independent index set. In other words, each dataset slice (i.e. segment) is a complete and independent unit. The Data Fabric logically connects these segments based on a particular query plan. Each one of these segments are represented by three Data Edge file types. These types will be labeled M, S, and L, where M is the manifest of data about the data and logically connects multiple S and L pairs. The M file is relatively small, where S and L pairs are in the range of 25 MB to 75 MB. This sizing is relative and based on the data source indexed, as well as, optimal S3 access.

As a result, for the **CHAOS**SEARCH service to index 1 TB of ELB logs, it creates 150 GB of Data Edge indexing equating to 3K in M, S, L files where S3 would be written to roughly 6K times. If the minimum S3 performance and limits are assumed, 100 Request Per Second with 50 MB per second (see Amazon S3 Announces Increased Request Rate Performance), today's S3 is more than fast enough to handle 150 GB of indexed data. Therefore, a 10-node fabric can easily index 1 TB of ELB logs data in about 1 hour.

Finally, any aspect of the Data Edge indexing can be changed dynamically. For instance, the size of S and L pairs can be increased, decreased, and/or split or merged to produce optimal S3 access. For instance, the Data Fabric will detect whether two or more index sets have common or related information, and will merge them together to reduce duplication, entropy, and number of S and L pairs. Merging is dependent on the data source indexed but can greatly reduce the size of the overall logical index as well as the number of S3 requests needed to materialize the original dataset.

Data Edge Materialization

Materialization (think search and/or query) of a dataset based on Data Edge indexing is extremely straightforward and highly optimized. For each logical index there is a "root" M file that describes the topology of the dataset. M files can be summarized and reference other "leaf" M files which typically represent raw data up to 100 GB. These M files are purposely designed to scope requests to resolve a particular search and/or query. In other words, determining whether an S and L pair does "not" have information related to a specific request is core to its existence.

This greatly reduces the number of S3 access points (think seeks) that the Data Fabric needs to materialize a result. In principle, one summarized M file could determine that a particular search/query result set was not in the logical index, and with one S3 access, resolve the request. But in practice, leaf M files will be utilized where 1 TB will have at least 10 to 100 M file accessed to determine whether part of an index set is required to materialize the result.

Therefore, only 1 to 10 accesses are needed to resolve a "nonexistent" request. And when determining if a symbol is within an S and L pair via a text search query or if a range is within a pair via a relational query, the M file can quickly scope these requests as well. It should also be noted, M files do not just to scope requests, but can actually resolve a particular search/query based on its statistical metadata.

The next step in scoping is the analysis (think query plan optimization) of a particular request to determine if an S and L pair is required. This is where S files (typically ⅓ size of L) is accessed to truly determine if the overall pair is required to materialize a result set. It is also important to understand that Data Edge scoping is driven via M to S to L existence optimizations. In other words, if an S file needs to be accessed, it has been determined that there is likely that the S and L pair are required to materialize. It is 100% deterministic on whether the L is required via S data.

Up to now, the focus has been around optimal accessing of S3. In other words, Data Edge has been optimized to find information efficiently. Data Edge internally orders information and can naturally reproduce the original data source order. However, for any relational query request where multi columns define the key spec (i.e. sort order), only the L file is required. The S file is only required on final materialization. Therefore, to sort a section of the 1 TB ELB dataset, the scoping will direct access to the pairs only required to resolve the request, where L files will first be ordered, grouped and/or joined and S for materialization. Data Edge also uniquely performs aggregations during this materialization phase. All of this greatly reduces seeks, network, and computation required.

What's Next

The **CHAOS**SEARCH service is backed by powerful new technology and architecture designed for multiple use cases and deployment options. Initially, we solve the inherent challenges of live and historical log and event analysis where existing solutions are cost prohibited. However, the platform has married storage and analytics, with a data lake philosophy, where analysis can be performed in-place, without the heavy lift, at an extremely low cost. The possibilities are endless!

For more information on **CHAOS**SEARCH, contact us at **chaossearch.io/contact**.

CHAOSSEARCH

About CHAOSSEARCH

CHAOSSEARCH is the creator of the first solution to make cloud storage searchable. The platform enables organizations to store, search, query, and visualize terabytes of data within their own Amazon S3 infrastructure with disruptive cost economics. **CHAOS**SEARCH gives customers a multi-tenant service model designed to accelerate innovation and drive business growth.

Founded in 2017, **CHAOS**SEARCH is a privately held, Boston, MA, company backed by .406 Ventures, Glasswing Ventures, and Stage 1 Ventures. For more information, visit us at chaossearch.io or follow us @CHAOSSEARCH.

CS-WP-02