Splunk® Data Fabric Search
Data Fabric Search 1.1.1

DFS overview

Generated: 4/30/2020 3:01 am

Copyright (c) 2020 Splunk Inc. All Rights Reserved
Data Fabric Search (DFS) is the new extended search platform that leverages the distributed processing power of compute clusters to broaden the scope and capability of Splunk Enterprise. Evolving and diverse data sources obscure visibility and makes it difficult to gather quick insights from data. Massive data stores also increase storage and processing costs. In short, analyzing data becomes harder when volume, variety, and the velocity of data grows at a massive scale.

DFS basically connects massive datasets from different data sources like mainframe computers, databases, IoT devices, and multiple Splunk deployments, which may have different storage and retention policies, into a single view. You can use DFS to search across multiple terabytes of data, billions of events without any performance impact, to gather Enterprise-wide insights into your data.

Traditionally, the Splunk platform runs searches from a single search head. The search query comes in through the search head and the results are returned back to the search head. The multiple reporting processors running on the search head in a single machine often create a performance bottleneck on the search head, which impacts the responsiveness of reporting, monitoring, and alert operations. A DFS job enhances search performance by distributing the search processing load to the compute cluster, so that processing and memory requirements do not cause a bottleneck at the search head. This data processing between the indexers, compute nodes, and search head optimizes the search process.

In short, the search head receives a DFS SPL search query. The search head only breaks up and defines the sequence and location of how the various components of the search will be implemented. The indexers pre-process and send the data to the DFS worker or compute cluster nodes. The compute cluster processes the search and sends the results back to the search head. The search head then applies relevant knowledge objects to the results and sends the final results back to the UI.

**Big data challenges addressed by DFS**

1. Searches may take too long to run.
   The following search may take around 90 minutes to run and return over a billion events.
   
   ```
   ... | stats count by city, org, ip | stats count
   ```
   
   High cardinality searches especially can generate millions of rows of results.

2. Memory restrictions may cause a search to fail.
   The following search may fail due to memory restrictions.
   
   ```
   ...| stats sum (duration) sum (price) by customer
   ```
   
   High cardinality searches especially can use extensive memory resources during event count estimation.

3. Inability to securely investigate data located in different regions.
   The following search may be a three step process that involves:
   
   1. running a search on `index = us`;
   2. running a search on `index = eu`;
   3. combining the two searches in a single report;

   ```
   index = us | stats count by cid | join type = inner left = L right = R where L cid=R cid [index = eu | stats count by cid]
   ```
   
   In contrast, a DFS search can use a single SPL query to search across multiple Splunk deployments or external federated providers, while leveraging existing knowledge objects and maintaining user access permissions. You do not need to move data across deployments to run a DFS search or unlock data to ensure users can access
only authorized datasets.

**DFS features**

DFS has the following features:

- **Big Data Analysis**: Analyzes and explores large amounts of data to search over a billion events within a single Splunk deployment with significant performance improvements. Additionally, DFS also allows you to perform high-cardinality searches where events have very uncommon or unique values. In fact, the higher the data cardinality, the higher the performance.

- **Federated Search**: Conducts searches and joins across multiple indexes on disparate Splunk deployments as seamlessly as if it was a single deployment.

Federated searches use an authorization model that enables the administrator to create service accounts for role-based user authentication across multiple Splunk deployments. A federated search head is a Splunk instance that handles search management functions, which directs search requests to federated providers that are remote Splunk Enterprise deployments. Federated searches provide the ability to correlate across a wider data fabric of multiple and disparate Splunk Enterprise deployments to access relevant datasets. The compute cluster applies the search pipeline to the results in a distributed manner. A remote search head is the Splunk Enterprise instance that resides on the remote Splunk deployment and conducts federated searches.

The following diagram illustrates the differences between a distributed search and a distributed search with a DFS compute cluster:
TLS is not enabled by default for data transport within a DFS deployment. For more information on securing your DFS deployment, see Secure a DFS deployment.

**Benefits of DFS**

In the last few years, there has been a tremendous boom in data generation which shows no signs of slowing down. DFS offers the following benefits when processing this “big data”:

**Scalability and complex joins**

DFS helps to aggregate complex datasets of billions of events to generate a quick report in minutes.

**Performance despite high data cardinality**

DFS can improve your ability to conduct high-cardinality searches on large volumes of data without compromising performance. If your dataset contains too many uncommon or unique values e.g. usernames, user IDs, email addresses etc. you can still search and gather insights from your data. In fact, the higher the cardinality of the data, the higher the performance of a DFS search.

**Longer time span analysis**

When you want to conduct searches over massive stretches of time to gather better insights into your data, DFS allows you to do that without impacting performance.

**Extensibility**

With federated search, you can join and search across multiple Splunk deployments seamlessly as if it was a single deployment.

**Role-based data isolation**

By using defined role capabilities, DFS can help you to ensure that data is not compromised across multiple deployments through restricted access to datasets and data sources.

DFS uses the Splunk DFS Manager app to automatically set up the compute cluster for your DFS deployment. To set up your compute cluster using Splunk DFS Manager, see Install Splunk DFS Manager.