xcalar

SOLUTION BRIEF

Device Usage and Failure Analytics for F500 Data Storage Vendor

AT A GLANCE

Challenges

- Completing projects using existing big data solutions required armies of programmers and consultants, thus reducing ROI and scope
- In-house SQL and Python experts were unable to move projects with other big data platforms
- Information extraction and insight discovery was complex and time-consuming, and resulted in gaps when handling unstructured data

Solution

- Deployed Xcalar as an end-to-end analytics pipeline – sourcing data, cleansing and transforming it, and delivering ad-hoc analytics to business analysts
- Used Xcalar Design to empower analysts to do their own data prep and analytics

Value

- Time to value reduced from 3 months to 4 days
- 10X increase in developer productivity
- 200X improvement in query performance

A leading computer hardware company has systems deployed around the world. To better understand how their products are being used, this company collects telemetry from most of these deployed systems. Data bundles arrive daily from several hundred thousand systems, resulting in petabytes of raw structured, semi-structured, and unstructured data to store, manage, and analyze. Xcalar partnered with this company's Data Analytics team to improve efficiency and completeness of the data workflow.

Challenges

The size and complexity of data bundles required a complex tool chain to process and analyze. A typical bundle consists of more than 100 sections, each represented by a single file. These files come in a wide variety of formats, including XML, Excel, binary, logs, text-based tables, and free-form text, with some containing more than one character set. File sizes range from a few kilobytes to over one gigabyte. Field counts per file type vary from a handful to over 500. Formats vary widely across system types, product models, and software versions. It has been a challenge to gain insights from the data, determine root-causes of system problems, and predict future problems, for the following reasons:

- Required Tool Expertise: Deriving insights required specialized knowledge with each tool in the chain, involving developers in different groups and geographies.
- Complex, Rapidly Changing Bundles: Raw data bundles were pre-processed to intermediate on-disk tables, using a complex set of parsers thousands of lines long. Few people understood this transformation or how to maintain it, but format changes and enhancements occur frequently, thereby straining parser maintainers.
- No Process Control: There was no mechanism to measure process accuracy, completeness, and ensure record uniqueness. There was no visible lineage back to the original data.
- Little Code Reusability: To gain insights from these complex source files, traditional analysis tools required programming to discover correlations, analyze trends, or predict problems, failures, or outages, but provide no structure for reuse. Without a way to perform these actions within a modular framework, the iteration and debug cycles became lengthy and inefficient.

Solution

The solution leverages many features provided by the Xcalar Design visual interface, including shared datasets and UDFs (User-Defined Functions), custom Python parsers, profiling and statistical analysis, data lineage, and auditability. Xcalar Compute Engine provides the scale and performance to model dataflows on multiple datasets with hundreds of millions of rows. In addition, Xcalar Compute Engine operationalized these dataflows for entire petabyte datasets.

Parsing By Data Bundle Section

Users created datasets by importing a set of files on an NFS server that contain a specific section in each data bundle. Users started with one dataset, then created additional datasets containing the information they find they need during modeling, and derived insights by joining datasets using a unique data bundle identifier as the join key. Xcalar natively imported bundle sections in open formats, such as XML, Excel, CSV, JSON, Parquet, or text. Users developed short Python import UDFs to parse custom file formats, such as tables-as-text or key-value pairs interspersed with unrelated text. In this fashion, unstructured data can be either imported as a block or refined into semi-structured or structured data. Because parsing is performed on one data bundle section at a time, parser code is modular, simple, and reusable. The original data files remain unchanged throughout; they are simply referenced as needed. No intermediate tables were needed to be written to disk, due to Xcalar's True Data In Place™ technology. Parser code was modified or updated directly in Xcalar Design, as bundle section formats evolved over time.

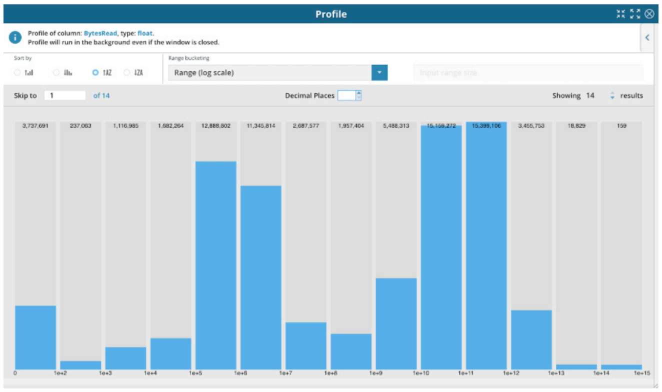
Visual Tools for Data Profiling, Exploration, and Validation

After users created a table from a dataset, they used Xcalar Design's visual design tools to understand and validate the data. Users applied profile operations on various columns in a table to confirm that each column contained valid content, and applied map or filter operations to weed out other integrity constraint violations. In addition, they created erroneous rows tables and filter complement tables to assess records failing integrity tests. Because Xcalar Design uses a powerful scale-out compute engine for modeling, users could play with tables of hundreds of millions of rows, and quickly validate, explore, and transform weeks' or months' worth of bundle data interactively. Some common data errors that users have discovered include:

- Duplicate, incomplete, or missing records
- Misspelled or inconsistent labels
- Inconsistencies between fields within certain records
- Non-unique key fields, such as serial numbers
- Inclusion of internal testing system data in production data analysis

Because data lineage to the original raw files is preserved, it is easy to determine the root-cause of data errors and resolve them.

Figure 1: shows an example Profile graph. The graph shows the distribution of the number of bytes read from a large number of disks, using a log scale.



Profile

Profile of column: BytesRead, type: float.
Profile will run in the background even if the window is closed.

Sort by: ○ ○ ● 1AZ ○

Range bucketing: Range (log scale)

Input range size

Skip to 1 of 14

Decimal Places

Showing 14 results

The combination of these visual modeling capabilities resulted in dramatic productivity improvements and insights at a level that was previously not possible with the company's existing analytics toolset.

Visual Tools Turn Focus To Custom Code

Because Xcalar's visual modeling tools empowered users to apply relational operations to tables created from validated bundle sections without code, developers were able to dedicate more time to applying advanced analytic techniques, such as predictive analytics and machine learning. Specific applications of advanced analytics are, as follows:

- Predictive analytics: Developers use predictive analytics to predict system failures for each product, so they can stock sufficient repair parts.
- Machine learning: Developers use the integrated Jupyter Notebook to develop algorithms, which train a TensorFlow ML model on a representative data sample. The user pastes the Python code and trained model into a Xcalar UDF, and applies classification and scoring operations in parallel across all nodes and cores in the cluster.

Operationalizing Results Using Batch Dataflows

To apply the results of modeling to their operations, users create batch dataflows from their modeling operations in Xcalar Design, and schedule the batch dataflows to execute regularly. A prime example is their application of the time-series analysis to understand downgrades. They created a batch dataflow that pulls contact information for owners of systems on which software had been downgraded. When they operationalized it, they applied it to their entire dataset; this pulls the contact information for all users who downgraded within the past month. The company then contacted each customer to determine reasons for the downgrade, creating a huge bump in both product quality and customer relationships.

Key Xcalar Benefits

- Data Exploration and Visualization: Xcalar Design's visual modeling capabilities means that query language expertise is no longer required to develop complex dataflows.
- Data Lineage: Dataflows show lineage back to the original data bundles. This makes it easy to validate transformation steps taken during analysis and trace problems back to the original data.
- Simplicity: Users perform all steps, such as data import, data preparation, cleansing, and analytics, within a single visual tool, freeing developers to do advanced analytics work.
- Data Access: Xcalar parses custom data source formats in parallel using small, modular Python import UDFs. Users refine import UDFs in Xcalar Design, then share them with other users.
- Parameterization: Once the model is developed, a batch dataflow can be run against data from various date ranges. For example, a user can build a dataflow model on one month's worth of data, then parameterize it to process the data for the entire year.
- Performance and Scalability: Real-time visual analysis on hundreds of millions of records in real time is enabled by a scale-out architecture and minimal network data transfers.
- Machine Learning: The integrated tool Jupyter Notebook is well suited for training ML models. Once these models are developed, they can be applied using Xcalar on large datasets iteratively in parallel across all cores and nodes. For continuous cycle algorithmic development, confidence scoring and retraining models is performed in Jupyter Notebook, as well.

KEY FEATURES, PRODUCTS, AND SERVICES

Key Features

- Point-and-analyze any source data file – unstructured, structured and semi-structured
- Custom sources can be imported by sections as separate datasets
- Visual programming, SQL, and Python for coding development flexibility
- Integrated machine learning tools, including Tensorflow

Products

- Xcalar Data Platform Premium Edition
- Xcalar Design Enterprise Edition
- Xcalar Enterprise Manager

Services

- Product training
- Solution architecture and design
- Infrastructure setup, configuration, and monitoring in the AWS environment
- User Defined Function for data import and export
- Transformation design and implementation
- Data flow design and implementation
- Cluster sizing and performance tuning

About Xcalar

Xcalar is an open and extensible analytics platform for the complete analytics pipeline that includes data quality, virtual data warehousing, data science, and workload operationalization. Users interactively build dataflows using visual design, SQL, and structured programming, and execute them at petabyte scale on unstructured, structured, and semi-structured data. Xcalar's enterprise-grade software scales to hundreds of nodes and thousands of users for both cloud and on-premises deployments. Its patented technologies deliver actionable insights with simplicity, speed, and scale.

Xcalar, Inc. 3031 Tisch Way Suite 450 San Jose CA 95128 USA Tel 408-471-1711 www.xcalar.com

06/18