xcalar

解决方案简介

面向 F500 数据存储供应商的设备使用情况和故障分析

概览

挑战

- 使用现有的大数据解决方案完成项目需要大批的程序员和顾问，因而会降低投资回报率并缩小范围
- 内部 SQL 和 Python 专家无法通过其他大数据平台迁移项目
- 提取信息和发现洞察信息的工作复杂而耗时，在处理非结构化数据时形成了空缺

解决方案

- 部署 Xcalar 作为端到端分析管道 – 搜寻、清理和转换数据，并为业务分析师提供特别分析
- 使用 Xcalar Design，让分析师能够完成自己的数据准备和分析

价值

- 价值实现时间从 3 个月缩短为 4 天
- 开发人员生产力提高 10 倍
- 查询性能提高 200 倍

一家领先的计算机硬件公司在世界各地部署了系统。为了更好地了解其产品的使用方式，该公司从部署的大多数系统中收集遥测数据。他们每天都会收到来自数十万个系统的数据包，这产生了数 PB 的结构化、半结构化和非结构化原始数据，需要进行存储、管理和分析。Xcalar 与该公司的数据分析团队进行了合作，以提高数据工作流的效率和完整性。

挑战

由于数据包的大小和复杂性，需要复杂的工具链才能进行处理和分析。一个典型的数据包包含 100 多个部分，每个部分都由单个文件表示。这些文件有多种格式，包括 XML、Excel、二进制文件、日志、基于文本的表和自由格式文本，其中一些包含多个字符集。文件大小从几千字节到超过千兆字节不等。每种文件类型的字段数从少量到超过 500 不等。系统类型、产品型号和软件版本不同，格式也大不相同。由于以下原因，要从数据中获取洞察信息，确定系统问题的根本原因并预测未来的问题一直都是困难重重：

- 需要工具方面的专业知识：要获取洞察信息，需要关于链条中每个工具的专业知识，这涉及不同团队和地理位置的开发人员。
- 数据包复杂且瞬息万变：原始数据包被预处理到中间磁盘表，使用的是一组复杂的长达数千行的解析器。很少有人了解这种转换或其维护方法，但格式更改和增强却经常发生，从而使解析器维护人员倍感压力。
- 缺乏流程控制：没有相应的机制来衡量流程的准确性和完整性，也无法确保记录的唯一性。无法将沿袭路径直观地回溯到原始数据。
- 代码不可重用：要从这些复杂的源文件中获得洞察信息，传统的分析工具需要进行编程才能发现相关性、分析趋势或预测问题、故障或中断，但不能提供可重用的结构。如果没有办法在模块化框架内执行这些操作，则迭代和调试周期会变得冗长而低效。

解决方案

解决方案是利用 Xcalar Design 可视化界面提供的众多功能，包括共享数据集和 UDF（用户自定义函数）、自定义 Python 解析器、性能分析和统计分析、IMD（插入/修改/删除）、数据沿袭和可审计性。Xcalar 计算引擎具有在多个包含数亿行内容的数据集上对数据流建模的规模和性能。此外，Xcalar 计算引擎可针对所有 PB 级数据集或特定的源数据日期范围运行这些数据流。

按数据包的部分解析

用户通过在 NFS 服务器上导入一组文件来创建数据集，这些文件包含每个数据包中的特定部分。用户先创建一个数据集，然后创建包含他们在建模过程中发现需要的信息的其他数据集，并通过使用唯一的数据包标识符作为连接键连接数据集来获得洞察信息。Xcalar 以开放格式在本地导入数据包部分，例如 XML、Excel、CSV、JSON、Parquet 或文本。用户开发了简短的 Python 导入 UDF 来解析自定义文件格式，例如表格文本或键值对，其中散布着不相关的文本。通过这种方式，非结构化数据既可以作为块导入，也可以细化为半结构化数据或结构化数据。由于解析一次只在一个数据包部分上执行，因此解析器代码是模块化、简单且可重用的。原始数据文件始终保持不变；它们只在需要时才被引用。由于 Xcalar 的 True Data In Place™ 技术，用户无需将中间表写入磁盘。随着数据包部分的格式随时间推移而改变，解析器代码直接在 Xcalar Design 中修改或更新。

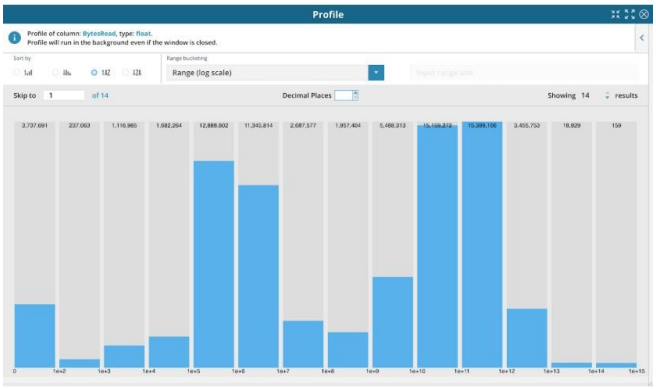
用于数据剖析、探索和验证的可视化工具

用户从数据集创建表后，再使用 Xcalar Design 的可视化设计工具来理解和验证数据。用户对表中的各个列应用配置文件操作，以确认每个列都包含有效内容，并应用映射或筛选操作以清除其他违反完整性约束的行为。此外，他们还创建了错误行表和筛选补充表，以评估记录是否能通过完整性测试。由于 Xcalar Design 使用强大的横向扩展计算引擎进行建模，因此用户可以使用包含数亿行内容的表，并以交互方式快速验证、探索和转换数周或数月的数据包数据。用户已发现的一些常见数据错误包括：

- 记录重复、不完整或缺失
- 标签拼写错误或不一致
- 某些记录中字段之间不一致
- 系统带有非唯一的序列号
- 系统用于内部测试，无法进行分析

由于保留了最初原始文件的数据沿袭，因此很容易确定数据错误的根本原因并加以解决。

表 1：显示了一个轮廓图示例。该图使用对数刻度显示了大量磁盘的字节数读数的分布情况。



这些可视化建模功能组合在一起显著提高了生产力，并获得了以前使用公司现有分析工具集无法获得的洞察信息。

可视化工具将关注重点转向自定义代码

借助 Xcalar 的可视化建模工具，用户无需代码即可将关系操作应用于从经过验证的数据包部分创建的表，因此开发人员可以将更多时间用于应用高级分析技术，例如时间序列分析、预测分析和机器学习。高级分析的具体应用如下：

- 时间序列分析：用户查看简单时间点数据聚合的时间窗口，以通过 IMD 跟踪软件版本升级和降级。然后，他们将该信息与其他系统特征相结合，以关联分析管理员在特定时间升级或降级某些系统的原因。以后，应用其他高级分析技术的现象将不再罕见。
- 预测分析：开发人员使用预测分析来预测每种产品的系统故障，这样他们就能存储足够的维修部件。
- 机器学习：开发人员使用集成的 Jupyter Notebook 来开发算法，这些算法通过有代表性的数据样本训练 TensorFlow ML 模型。用户将 Python 代码和训练好的模型粘贴到 Xcalar UDF 中，并在集群中所有节点和核心上并行应用分类和评分操作。

使用批处理数据流实施结果

要将建模结果应用到操作中，用户可以在 Xcalar Design 中从其建模操作创建批处理数据流，并安排定期执行这些数据流。最好的例子就是，他们应用时间序列分析来理解降级行为。他们创建了一个批处理数据流，用以提取软件被降级的系统的所有者联系信息。当他们实施该数据流时，就将其应用到了整个数据集；这会提取在过去一个月内进行了降级的所有用户的联系信息。然后，该公司会与每位客户联系，以确定降级的原因，从而大幅改善产品质量和客户关系。

Xcalar 的主要优势

- 数据探索和可视化：Xcalar Design 的可视化建模功能意味着，不再需要查询语言专业知识即可开发复杂的数据流。
- 数据沿袭：数据流可以显示沿袭路径，一直回溯到原始数据包。这样就可以轻松验证分析过程中所采用的转换步骤，并将问题追溯回原始数据。
- 简便性：用户可以在单个可视化工具中执行所有步骤，例如数据导入、数据准备、清理和分析，从而使开发人员有时间执行高级分析工作。
- 数据存取：Xcalar 使用小型的模块化 Python 导入 UDF 并行解析自定义数据源格式。用户可在 Xcalar Design 中优化导入 UDF，然后与其他用户共享。
- IMD：Xcalar Design 中的 IMD 扩展是一款功能强大的时间序列分析工具，用于分析不同时间的性能和系统利用率，并确定转换点，如软件升级和降级。
- 参数化：模型开发好后，就可以针对不同日期范围的数据运行批处理数据流了。例如，用户可以为一个月的数据构建数据流模型，然后对其进行参数化，以处理整年的数据。
- 性能和可扩展性：通过横向扩展架构和最小的网络数据传输，实现对数亿条记录的实时可视化分析。
- 机器学习：集成的工具 Jupyter Notebook 非常适合训练 ML 模型。模型开发好后，就可以通过 Xcalar 在大型数据集上对所有核心和节点反复地并行应用。对于连续循环算法开发，还可以在 Jupyter Notebook 中执行信心评分并再训练模型。

主要特性、产品和服务

主要特性

- 只需点击即可分析所有源数据文件 – 非结构化、结构化和半结构化
- 自定义源可以按部分导入为单独的数据集
- 可使用可视化编程、SQL 和 Python 进行灵活的代码开发
- 集成了机器学习工具，包括 Tensorflow

产品

- Xcalar 数据平台高级版
- Xcalar Design 企业版
- Xcalar 企业管理器

服务

- 产品培训
- 解决方案架构和设计
- 在 AWS 环境中设置、配置和监控基础架构
- 通过用户自定义函数导入和导出数据
- 转型设计与实施
- 数据流设计和实施
- 集群大小调整和性能优化

关于 Xcalar

Xcalar 是一个可扩展的开放性分析平台，适用于完整的分析管道，包括数据质量、虚拟数据仓库、数据科学和运维。用户可使用可视化编程、SQL 和结构化编程以交互方式构建数据流，并对非结构化、结构化和半结构化数据以 PB 级别执行这些数据流。Xcalar 的企业级软件可扩展到数百个节点和数千个用户，既可进行云部署也可进行本地部署。凭借其专利技术，用户能轻松、快速、大规模地获得可操作的见解。

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

06/18