SQL GR TERADATA ASTER PAPER pdf

1: Teradata Aster Analytics Foundation User Guide

The TeradataÂ® AsterÂ® Discovery Platform is the industry's first next-generation, integrated discovery platform that provides powerful insights from big data.

Scaling the processing of graph data such as social network data, telecom data, and public health disease outbreak data is notoriously challenging. Unlike set-oriented relational database processing which is fairly straightforward to parallelize, graph processing typically requires traversing through paths in a graph. Given the high branching factors of most real-world graphs, this results in non-local access patterns, and thus a high dependence on vertices and edges being stored or processed by remote processors. The reliance on remote data makes parallel processing much more difficult, as each processor may have to wait for arbitrary lengths of time for remote data. However, by eschewing asynchronous communication models that can lead to deadlock and other types of race conditions that make programming scalable algorithms extremely complex, BSP makes it much easier for programmers to reason about the semantics of their graph algorithms. In each super-step, each vertex, v, in the graph can apply an arbitrary function on its own local state, based on messages sent to v from any other vertex in the graph in the previous super-step. The output of this function is a new state for v and also potentially new states for its outgoing edges , and a set of messages that can be sent to any arbitrary vertex in the graph that will be processed in the next super-step. In other words, processing proceeds in a BSP graph system via an alternating series of completely local function applications done in parallel on each vertex in the graph, interspersed with an arbitrary amount of message passing between vertexes. Thus, the combination of programmer friendliness, along with the relatively small cost of the synchronization barriers, has led to the BSP model for graph processing to gain enormous popularity over the four years since the Pregel paper was published. This greatly increases both the power and efficiency of the system, which the rest of this blog post will discuss. Furthermore, it contains various dimension tables containing customer address and billing information, router location information, etc. For most of what the company wants to do with this data e. Dimension tables are joined with the fact table as needed, and some may even be pre-joined with the fact table if a column-oriented layout is used since large, wide tables are not problematic for column-stores. However CDR data is also a gold-mine for doing customer churn analysis because it contains information about real connections between people who is calling whom, which numbers are popular, who are the bridges between different communities of customers, etc. For this type of analysis, it is helpful to think of CDR data as a graph, where each customer or phone number is a vertex in the graph, and there are edges in the graph for each CDR record corresponding to calls and SMS texting between two customers phone numbers. For example, if many important people call Mr. Measuring the eigen centrality of a vertex is necessarily an iterative algorithm in each iteration we get a new estimate of the importance of each vertex and can use this to refine the quality values of each outgoing edge in order to create a new importance estimate for each vertex in the next iteration. It calculates the number of times any vertex is included in the shortest path between any two nodes in the graph. If this number is large, this means that the vertex often serves as a bridge between multiple clusters of nodes in the graph, and has a large amount of power in the control of information through the graph. In our example, these clusters of vertices are communities of people. Centrality of a Vertex Between Clusters of Vertices Since both betweenness and eigen centrality are complex, iterative algorithms on potentially large graphs, BSP graph engines of the type discussed above are a great fit for performing these calculations. Trying to perform the same operations in a standard relational database system is close to impossible each iteration would require a self-join of the table containing all the edges in the graph; but without knowing in advance how many iterations will be required, it cannot be determined how many joins to include in the SQL query. Furthermore, even if the required number of iterations could somehow be deduced in advance, trying to write these complicated graph algorithms in SQL would be highly unpleasant and nearly impossible for other humans to decipher. Therefore, our example telecom company is a little stuck on one hand a relational database system is a great fit for most of what needs to be done with the data billing, fault analysis, stat generation, etc. Since traditional database systems do not process graph algorithms well, the telecom company may be tempted to put the entire dataset in the BSP engine. Unfortunately, while graph engines are great for running graph algorithms, they are not optimized for traditional relational analysis, where standard star-schemas and table-oriented especially column-oriented analysis are known to perform extremely well. Therefore, they will see major reductions in performance on their non-graph-oriented queries, and will further have decreased options analytical tools to use in conjunction with their data, since many third-party tools are designed to interface only with relational database systems. Another option is to simply duplicate the same CDR data in two different systems one relational system for doing standard analysis and one BSP graph system devoted to analyzing customer churn. Unfortunately, this option is also suboptimal. Aside from the problem of duplicating data in two systems and the additional associated hardware, administrative, and maintenance costs , there are also a class of queries that require both relational and graph processing techniques. The segmenting of the original dataset by these dimensions to get the new graph is exactly what relational database systems are designed for, while the calculation of betweenness metrics are what BSP graph systems are designed for. Performing this query in two separate systems would require the relational database system to perform the first part of the query, then export the new graph before it gets parsed and reloaded into the BSP system for the second part of the query. If further relational processing needs to be performed on betweenness metrics, another export and reload back into the relational database system will have to occur. The entire query can be expressed as a single request, and processed entirely by one system. For example, given the following two tables with the relevant attributes listed below: Since this post is already quite long, we will not discuss this optimization technique here, but I will likely describe it in detail in a future post. In summary, graph analysis is an increasingly important technique that data scientists will need to apply to many real-world datasets. BSP graph engines are extremely well suited to performing a large class of graph analysis queries; however, they do not replace traditional relational database systems. Since modern workloads will contain both relational and graph queries, including some queries that require both relational and graph techniques within the same query, a two system approach will lead to headaches, slowdowns, and a general lack of optimality. Thus, I believe that many relational database systems will eventually incorporate a BSP execution engine to enable hybrid workloads and analysis. Teradata Aster is at the forefront of this trend. Daniel Abadi is an Associate Professor at Yale University, founder of Hadapt, and a Teradata employee following the recent acquisition. He does research primarily in database system architecture and implementation. He received a Ph. He is best known for his research in column-store database systems the C-Store project, which was commercialized by Vertica , high performance transactional systems the H-Store project, commercialized by VoltDB , and Hadapt acquired by Teradata.

SQL GR TERADATA ASTER PAPER pdf

2: Aster | Teradata Downloads

Teradata on Tuesday introduced a major upgrade of its Teradata Aster Discovery Platform with a twist: a new SQL-based graph-analysis engine to go along with the platform's existing SQL query and SQL-based MapReduce capabilities.

If you ask questions or solve day to day business problems with a language like SQL, you are using a declarative language. In relational world, structure is provided by tables and logic is provided by joins, functions, casts, etc. Data is structured, so logic can be imposed on it. You code in Python, R, Java or Scala to unravel the structures and perform analysis. You can dig through JSON structures, parse binary streams etc. Data cannot be structured that easily and only procerdural code or a programming language can wrangle it case by case. The interface is procedural, but the access method continues to be declarative. Trade-offs between Procedural and Declarative interfaces - Why some prefer one over the other: It is unfair to over generalize which set of users prefer which, but there is a subtle pattern. Feel free to comment at the end if you disagree. Declarative interfaces like SQL are meant to simplify the common tasks with a standardized abstraction. A lot of business analysts use SQL. As more machine learning and analytics becomes well understood and mainstream, they also get embedded and available in SQL as a standardized way of accessing. You can see how invoking graph and pathing analytics is easier and intuitive through a native SQL declarative interface. The above is from the Postgres user guide. No SQL was involved here. What about Apache Spark? The primary access method is from a procedural language like Scala, R, Python, Java etc. See Scala code below that embeds SQL inside. So who would use which? Insights sharing tools like Teradata Aster App Center, R Shiny also makes sure that regardless of the method used, the workflow is packaged to be consumed to the end user with a browser interface

SQL GR TERADATA ASTER PAPER pdf

3: SQL-GR | Teradata Downloads

Teradata Â® Aster Analytics is the industry's first multi-genre advanced analytics solution that provides powerful, high-impact insights on all data types of any volume.

Teradata Aster 6 has been preannounced beta in Q4, general release in Q1 The general architectural idea is: There are multiple data stores, the first two of which are: The classic Aster relational data store. Generic analytic SQL, as Aster has had all along. The Aster parser and optimizer accept glorified SQL, and work across all the engines combined. Just to be clear: The most dramatic immediate additions are in the graph analytics area. BSP was thought of a long time ago, as a general-purpose computing model, but recently has come to the fore specifically for graph analytics. Think Pregel and Giraph, along with Teradata Aster. BSP has a kind of execution-graph metaphor, which is different from the graph data it helps analyze. Aster long ago talked of adding a graph data store, but has given up that plan; rather, it wants you to do graph analytics on data stored in tables or accessed through views in the usual way. Use cases suggested are a lot of marketing, plus anti-fraud. Within those functions, the core idea is: It can get reasonably complicated; e. Truth be told, these prebuilt functions sound pretty interesting. As for underpinnings â€" the idea behind BSP is: You have a computing job that is both iterative and parallel. You parallelize it among a bunch of logical vertices, which may or may not correspond to physical computing servers. At the end of a superstep, each vertex can send messages to other vertices. Hopefully, various problems with message latency and unreliability that arise in other models of parallel computing are obviated by BSP. So why use BSP for graph analytics? Real-world graphs have short average path lengths â€" Six Degrees of Separation and all that. Subgraph pattern matching was mentioned as an area that was not yet optimized for. But with all those caveats, this still feels like a pretty interesting entry into the relationship analytics market.

4: Aster 6, graph analytics, and BSP | DBMS 2 : DataBase Management System Services

For example, the original Pregel paper has already received close to citations, and several new graph processing systems that adopt the Pregel BSP model have been recently developed, including Apache Giraph, Apache Hama, GPS, Mizan, and Teradata Aster's SQL-GR framework.

Like other graphing engines, SQL-GR is designed to allow customers to establish connections among various entities stored in the database, which could represent anything from a particular customer to a bank account to a fine French Bordeaux, and are often stored in unstructured or semi-structured formats. Graphing engines in general have become popular because they provide another way to make sense out of massive amounts of unstructured and semi-structured data, like emails, Web logs, machine data, and sensor data. Graphing engines are good at identifying patterns in the data that might otherwise be difficult to spot using traditional SQL or even MapReduce. The company calls this its SNAP framework. Prior to this release, Aster had a purely relational data store. With version 6, it added a new file-based data store to keep the semi-structured and unstructured data. Using Aster 6, customers today can analyze that email-based data through the comfort of SQL. However, the capabilities will be improved with coming releases. One potential use of this functionality is identifying churn patterns in the telecommunications industry. Then the company can perform statistical analysis to pull up messaging statistics, and see how active they are. Then they can pass the results over to a MapReduce function, to perform sentiment analysis to see whether any of the people have been calling into the call center to complain about bad service. But what about salad and condiments? Does a certain selection of wine determine whether a customer is more likely to choose a certain crouton? The graphing engine could help a grocery store direct certain types of users to higher margin products. These types of challenge and solutions are fairly common in the big data analytics world. The advantage that Teradata provides with its Aster platform is the capability to access these rich analytics from plain old SQL. The MapReduce engine does its thing and the executor then takes the result and puts it into graph engine. This is all happening very transparent to the user, through the SNAP architecture. Aster 6 will be entering beta later this year, with GA expected in Aster can run on the same cluster where the Teradata data warehouse and Hadoop reside. Aster and Teradata can co-exist on any number of nodes in the cluster, up to a maximum of 18 nodesâ€”making for a very rich, very powerful big data offering.

5: sql | Teradata Downloads

Aster Database: In-Database Analytics with R. In this white paper, learn how Aster marries the power of standard SQL development with existing programming languages and new frameworks for analytic processing.

6: Graph Processing Inside an Analytic DBMS | Teradata Downloads

Teradata has today announced Teradata Aster R. This is designed to allow you to run R "in-database" on Aster's MPP architecture - to run open source R at scale. This is designed to allow you to run R "in-database" on Aster's MPP architecture - to run open source R at scale.

7: Teradata Aster gets graph database, HDFS-compatible file store | ZDNet

Teradata Studio Express now supports Aster database connectivity. Teradata Studio Express is an information discovery tool for retrieving and displaying data from your Aster database systems. It can be run on multiple operating system platforms, such as Windows, Linux, and Mac OSX.

8: Teradata Adds Graph Engine to the Data Discovery Mix with Aster 6

SQL GR TERADATA ASTER PAPER pdf

Aster Development Environment (ADE) is a development environment that lets you write, test, and deploy your Teradata Aster SQL-MapReduceÂ®, Teradata Aster SQL-GRâ„¢, Aggregate, and Scalar functions on an Aster instance.

9: Related Documentation

What can we help you find? Contact Us.

Community and a perspective Becoming A Master Student Concise 11th Edition Plus E-learning Companion Plus 2006-2007 Student Success P Real life applications of soft computing Basic Composition The Competency Casebook Setting Up an Out-of-school Club First Steps to Jesus-Blk Pearl You Decide! 2005 Edition (2nd Edition) Keepers of Florida lighthouses, 1820-1939 Italian film posters Managerial cost accounting and analysis History of public relations in india 20 Hands-On Activities For Learning Idioms The Masters Healing Presence Bible (Bible Av) Taming the Taildragger Sketch of the life and public services of Gen. Lewis Cass. She friendzoned my love Sweet bells jangled Essential of Vietnams business law Extrapolation methods theory and practice Modeling monetary economies solutions Editors that can rotate Pocket computer programs Lease of certain public property. Adventure Six: Lets Play With Information Dealing with an OSHA inspection and citation Michael G. Murphy How stress affects your health Suffering and Healing in Our Day (Proceedings of the Theology Institute of Villanova University) Users manual for the medical outcomes study (MOS core measures of health-related quality of life Learning theories for teachers Tamilnadu voter list 2017 Analytical lingustic concordance to the book of Isaiah New Shops Boutiques James sa corey cibola burn Journey for the planet The way to concrete realization. Momentum, Energy and Mass Transfer (McGraw-Hill chemical engineering series) Epson, Epson, read all about it! Socialist relations of production: lechery for labor Nurse, P. One rejection too many.