

ConcertedProposal

Abstract

Concerted is an in memory write less read more engine aimed to provide extreme read performance with very high degree of concurrency and scalability and focus on minimizing own resource footprint.

Proposal

Concerted is built on the principal that a new type of workload is dominating the scene and is now needed to be supported. These are the large data set analytical workloads being analyzed or used on large clusters or high power machines. Large analytical workloads depend on the ability to query large data sets efficiently and in high concurrency while maintaining semantics such as immediate consistency. An in memory engine designed to support extreme read queries while providing support for aggregation through various features (such as multidimensional representation of tuples) will accelerate many usecases around large scale analytics.

Concerted believes that best understanding of user application lies with user application developer. The need for massive read scaling should be on demand and should be flexible to the level that user can decide as to which representation and access of data suits his/her current requirements. Hence, Concerted is not built in a traditional client/server model. Concerted provides users with an API which can be used to load, read, update and delete data. User chooses which data structure has to be used for his current requirements. All API access is covered by Concerted's internal systems like lock manager, transaction manager and cache manager which ensure that reads scale to high level in every API call.

Concerted is a Do It Yourself in memory platform for making in memory supporting engines. The use case we think of is supporting big data warehouses like Hive, but there are endless use cases for a custom, highly scalable in memory platform.

The goal of this proposal is to leverage an existing code base available on Github and licensed under the Apache License 2.0 to build a community around the project. Currently the community consists of existing hackers of Concerted as well as people who have been following and associated with the project since a while as well as database experts who are excited about building a project like this. We are hoping that entering into Apache would help us attract more contributors as well as connect with existing big data projects like Apache Hive, Apache HAWQ, Apache Storm, Apache Tajo, Apache Spark, Apache Geode to leverage their community base while assisting in their use cases with Concerted. We had a discussion with founders of Apache Tajo and they showed interest in using Concerted for some of their use cases.

Background

Relational databases were built with the cost of physical memory in mind. The cost is no longer very relevant and physical memory is now available on demand. Another driving factor behind Concerted is that there is a paradigm shift with big data coming into picture. Disk IO speeds are more of a bottleneck than ever before. Combining the read dominance of analytical workload with the speed of in memory structures, Concerted fits the current scene. Also, supporting OLAP workloads with in memory support for faster read constant queries and joins will be useful.

Rationale

As explained above, large analytical workloads need an in memory lightweight engine which supports massive read concurrency, ground level support for aggregations and analytics, extreme scalability and high read performance, along with the engine being very light itself. Concerted aims to solve these needs. Concerted is designed and built with three goals as objectives:

Performance

To provide high performance access to data from a large number of rows, Concerted uses efficient representation and in memory indexing of data coupled with high performance transactions, custom transactions and lightweight locking and lockless techniques and an intelligent locking manager.

Scalability

Concerted is built with extreme concurrency and scalability in mind.

Efficiency

Concerted aims to give expected performance under vast variety of workloads and aims to have as low footprint as possible.

Initial Goals

The initial goal is to leverage an existing code base and invest in building a community around the project. We anticipate a lot of initial restructuring of the existing code so that it becomes easier to include new contributors and minimize ramp up time. We plan to approach this refactoring in a fully transparent, community-driven way thus starting to practice the "Apache Way" governance model from the get go.

Various contributors are getting individual changes into branches in github repository and our initial major goal will be to merge in all those changes in master repository.

Current Status

Concerted is currently under restructuring to suit the needs of an open source project. Current source is available at <https://github.com/atris/Concerted> (Please note that updated codebase is not yet present on github) Concerted is currently being licensed under Apache License 2.0. Most of the code base is implemented in C and C++ and has external dependencies listed later.

Meritocracy

We plan to drive the technical roadmap and implementation in a fully transparent, community-driven way soliciting feedback from all of the community members and building a consensus-driven approach to evolving the code base and the community itself. Users and new contributors will be treated with respect and welcomed. By participating in the community and providing quality patches/support that move the project forward, contributors will earn merit. They also will be encouraged to provide non-code contributions (documentation, events, community management, etc.) and will gain merit for doing so. Those with a proven support and quality track record will be encouraged to become committers.

Community

In memory is the new cutting edge thing and a new community around performance oriented systems and enhancing relational database performance by having complete in memory OLTP engines will greatly benefit performance. So we expect data warehousing projects and communities as well as projects and companies looking for high performance OLTP performance. In addition, Ingenium Data Systems is building products around Concerted and will have salaried developers contribute to the project as part of job responsibility.

Core Developers

Core developers are a diverse group of developers, many of which are very experienced in open source and the Apache Hadoop ecosystem. Specifically, Atri is an Apache Apex committer and Atri and Pavel are major contributors to PostgreSQL project. Atri is also committer for other open source projects.

- Amrish <amrishes AT ingeniumsys DOT com>
- Nupur S <nupurs AT ingeniumsys DOT com>
- Pavel Stehule <pavel DOT stehule AT gmail.com>
- Atri Sharma <atri AT apache DOT org>
- Nishith Singhal <nishsinghal AT gmail DOT com>
- Michael Down <michael AT dowuk DOT com>
- Vijayakumar Ramdoss <vijayakumar DOT ramdoss AT emc DOT com>
- Wang Albert <albertwang87 AT gmail DOT com>
- Hans-Jurgen Schonig <postgres AT cybertec DOT at>
- Kris Popat <krispopat AT apache DOT org>
- Ayrtón Gomez <com DOT ayrtón AT gmail DOT com>

Alignment

Concerted will be helpful to systems like Tajo which can benefit with in memory structures optimized for heavy reads and joins (dimension tables). In addition Concerted will benefit projects looking for in memory relational database as a metadata store, which is the case for most of the Apache Big Data projects. We expect Apache HAWQ (incubating), Apache Hive, Apache Storm, Apache Tajo to be utilizing Concerted as a supporting engine. For eg, a data warehouse built on HAWQ, Hive or Tajo can utilize Concerted as an in memory engine for querying and joining dimensional tables.

Known Risks

Orphaned Products

Most of the code is developed by a small group of core developers and this may be a risk for orphaned product. However, the code base is simple as compared to other open source projects and the interest level in Concerted has risen exponentially over the years with many computer professionals expressing interest in the project and doing some use cases of the same. Specifically, there were some projects done around Concerted in IIIT, Noida (an engineering school) and Wang is a student in Lehigh University who has been following Concerted's progress over many years. The core developers are aligned with this project and since the code base is simple, future committers will have a quick ramp up and the risk shall be mitigated. Besides, Ingenium Data Systems is launching a product based on Concerted and will be having all its salaried developers contribute to Concerted as a part of their job functions.

Inexperience with Open Source

Most of the initial committers have experience working on open source projects. In particular, Atri is an active member of many open source projects.

Homogeneous Developers

Although initial core developers were based out of India, community now consists of computer professionals from various parts of the world hence diversity should not be an issue. In addition, we will be documenting internals of the project in public facing documents and it shall allow more contributors to join in.

Reliance on Salaried Developers

It is expected that Concerted development will occur on both salaried time and on volunteer time. Nupur and Amrish belong to Ingenium and are committed to building this project along with their team. Atri, as the originator of this project, will be actively working on the project and is now pushing Concerted into major data warehousing projects, since he is involved in architecture of data platforms. Developers are expected to be contributing in their volunteer time. In addition, we will be working with various open source projects which will be benefited by Concerted and will be involving those communities into Concerted's development as well. For eg, Apache Tajo has shown interest and will be supporting development of the project.

Relationships with Other Apache Products

Concerted has some overlapping function with Apache Geode(Incubating). However, Geode is an in memory key value store whereas Concerted is a write less read many engine. Concerted will complement Geode and increase the use cases Geode can support with Concerted's help.

A major objective for Concerted is supporting OLAP workloads and data warehouses with in memory performance and highly performant reads and joins. Concerted will be collaborating with many open source projects such as Apache HAWQ (incubating), Apache Hive, Apache Tajo etc to support their OLAP workloads hence enabling them to support larger set of usecases with a better throughput. For eg, a star schema in Hive will benefit from having dimension tables in Concerted with highly efficient and scalable reads and joins will be very fast. Similar workload for Tajo.

Concerted will fit in many other use cases in Apache spectrum as well. For eg, Concerted can be used with Apache Geode for in memory aggregation indexing. Concerted can also be used with Apache Flink for streaming real time data into in memory, perform in memory aggregation and then performing batch processing for efficiency.

A Excessive Fascination with the Apache Brand

We believe that the "Apache Way" governance model will provide additional help to us in finding contributors and growing the community. The community and development process will make this project more stable and help establish ubiquitous APIs. In addition, Concerted is looking to support multiple Apache projects in their use cases and accelerate their performance while soliciting their support in development of the project. We will not be using Apache brand for excessive branding or with any commercial aspects of Concerted. Apache brand will primarily be used for community building.

Documentation

Public documents are currently in development and will be published soon.

Initial Source

The initial source is written in C++ and is heavily in development. It will be restructured and released publicly. We understand that there might be concerns around github source being developed by only a single person and development not happening after 2013. The source on github is only the source initially developed as an independent project hence the limitation. However, due to reason that project has been present on github for a while now, it has attracted attention and people have been using and developing it locally. For eg, Ingenium Data System took an interest in the project and locally developed it and used it in an upcoming product they are going to release soon. The project now wants to accumulate all independent development efforts and help attract people to grow the community and project. We are currently in process of updating github repository and making branches for all local development efforts.

Source and Intellectual Property Submission Plan

We intend the entire code base to be licensed under the Apache License, Version 2.0.

External Dependencies

Currently, Concerted only depends on g++ compiler and pthreads. pthreads will be replaced by Boost in next release.

Cryptography

N/A

Required Resources

Mailing List

*private@concerted.incubator.apache.org (moderated subscriptions)
*commits@concerted.incubator.apache.org
*dev@concerted.incubator.apache.org
*issues@concerted.incubator.apache.org

Git Repository

<https://git-wip-us.apache.org/repos/asf/incubator-concerted.git>

Issue Tracking

Jira Concerted (CONCERTED)

Other Resources

- Continuous Integration
 - Jenkins
- Wiki
 - cwiki.apache.org/confluence/display/CONCERTED

Initial Committers

- Roman Shaposhnik <rvs AT apache DOT org>
- Daniel Dai <daijy AT apache DOT org>
- Jake Farrell <jfarrell AT apache DOT org>
- Lars Hofhansl <larsh AT apache DOT org>
- Julian Hyde <jhyde AT apache DOT org>
- Chris Nauroth <cnauroth AT hortonworks DOT com>
- Pavel Stehule <pavel DOT stehule AT gmail.com>
- Amrish <amrishes AT ingeniumsys DOT com>
- Nupur S <nupurs AT ingeniumsys DOT com>
- Atri Sharma <atri AT apache DOT org>
- Nishith Singhal <nishsinghal AT gmail DOT com>
- Michael Down <michael AT dowuk DOT com>
- Vijayakumar Ramdoss <vijayakumar DOT ramdoss AT emc DOT com>
- Wang Albert <albertwang87 AT gmail DOT com>
- Hans-Jurgen Schonig <postgres AT cybertec DOT at>
- Kris Papat <krispopat AT apache DOT org>
- Ayrton Gomesz <com DOT ayrton AT gmail DOT com>

Affiliations

- Roman Shaposhnik (Pivotal)
- Daniel Dai (HortonWorks)
- Jake Farrell (Acquia)
- Lars Hofhansl (Salesforce)
- Julian Hyde (HortonWorks)
- Chris Nauroth (HortonWorks)
- Pavel Stehule (GoodData)
- Amrish (Ingenium Data Systems)
- Nupur S (Ingenium Data Systems)
- Atri Sharma (Barclays)
- Nishith Singhal (Wipro)
- Michael Down (Barclays)
- Vijayakumar Ramdoss (EMC)
- Wang Albert (Lehigh University)
- Hans- Jurgen Schonig (CyberTec)
- Kris Papat (CETIS LLP)
- Ayrton Gomesz (IQLabs)

The nominated mentors are employees of [HortonWorks](#), Acquia, and Salesforce.

- Daniel Dai (HortonWorks)
- Jake Farrell (Acquia)
- Lars Hofhansl (Salesforce)
- Julian Hyde (HortonWorks)
- Chris Nauroth (HortonWorks)

Sponsors

Champion

- Roman Shaposhnik (rvs AT apache DOT org)

Nominated Mentors

- Daniel Dai <daijy AT apache DOT org>
- Jake Farrell <jfarrell AT apache DOT org>
- Lars Hofhansl <larsh AT apache DOT org>
- Julian Hyde <jhyde AT apache DOT org>
- Chris Nauroth <cnauroth AT hortonworks DOT com>

Sponsoring Entity

Apache Incubator