AsterixDBProposal

Apache AsterixDB Proposal

Abstract

Apache AsterixDB is a scalable big data management system (BDMS) that provides storage, management, and query capabilities for large collections of semi-structured data.

Proposal

AsterixDB is a big data management system (BDMS) that makes it well-suited to needs such as web data warehousing and social data storage and analysis. Feature-wise, AsterixDB has:

- A NoSQL style data model (ADM) based on extending JSON with object database concepts.
- An expressive and declarative query language (AQL) for querying semi-structured data.
- A runtime query execution engine, Hyracks, for partitioned-parallel execution of query plans.
- Partitioned LSM-based data storage and indexing for efficient ingestion of newly arriving data.
- Support for querying and indexing external data (e.g., in HDFS) as well as data stored within AsterixDB.
- A rich set of primitive data types, including support for spatial, temporal, and textual data.
- Indexing options that include B+ trees, R trees, and inverted keyword index support.
- Basic transactional (concurrency and recovery) capabilities akin to those of a NoSQL store.

Background and Rationale

In the world of relational databases, the need to tackle data volumes that exceed the capabilities of a single server led to the development of “shared-nothing” parallel database systems several decades ago. These systems spread data over a cluster based on a partitioning strategy, such as hash partitioning, and queries are processed by employing partitioned-parallel divide-and-conquer techniques. Since these systems are fronted by a high-level, declarative language (SQL), their users are shielded from the complexities of parallel programming. Parallel database systems have been an extremely successful application of parallel computing, and quite a number of commercial products exist today.

In the distributed systems world, the Web brought a need to index and query its huge content. SQL and relational databases were not the answer, though shared-nothing clusters again emerged as the hardware platform of choice. Google developed the Google File System (GFS) and MapReduce programming model to allow programmers to store and process Big Data by writing a few user-defined functions. The MapReduce framework applies these functions in parallel to data instances in distributed files (map) and to sorted groups of instances sharing a common key (reduce) – not unlike the partitioned parallelism in parallel database systems. Apache's Hadoop MapReduce platform is the most prominent implementation of this paradigm for the rest of the Big Data community. On top of Hadoop and HDFS sit declarative languages like Pig and Hive that each compile down to Hadoop MapReduce jobs.

The big Web companies were also challenged by extreme user bases (100s of millions of users) and needed fast simple lookups and updates to very large keyed data sets like user profiles. SQL databases were deemed either too expensive or not scalable, so the
“NoSQL movement” was born. The ASF now has HBase and Cassandra, two popular key-value stores, in this space. MongoDB and Apache CouchDB are other open source alternatives (document stores).

It is evident from the rapidly growing popularity of "NoSQL" stores, as well as the strong demand for Big Data analytics engines today, that there is a strong (and growing!) need to store, process, *and*
query large volumes of semi-structured data in many application areas. Until very recently, developers have had to *choose* between using big data analytics engines like Apache Hive or Apache Spark, which can do complex query processing and analysis over HDFS-resident files, and flexible but low-function data stores like MongoDB or Apache HBase. (The Apache Phoenix project, http://phoenix.apache.org/, is a recent SQL-over-HBase effort that aims to bridge between these choices.)

AsterixDB is a highly scalable data management system that can store, index, and manage semi-structured data, e.g., much like MongoDB, but it also supports a full-power query language with the expressiveness of SQL (and more). Unlike analytics engines like Hive or Spark, it stores and manages data, so AsterixDB can exploit its knowledge of data partitioning and the availability of indexes to avoid always scanning data set(s) to process queries. Somewhat surprisingly, there is no open source parallel database system (relational or otherwise) available to developers today – AsterixDB aims to fill this need. Since Apache is where the majority of the today's most important Big Data technologies live, the ASF seems like the obvious home for a system like AsterixDB.

Current Status

The current version of AsterixDB was co-developed by a team of faculty, staff, and students at UC Irvine and UC Riverside. The project was initiated as a large NSF-sponsored project in 2009, the goal of which was to combine the best ideas from the parallel database world, the then new Hadoop world, and the semi-structured (e.g., XML/JSON) data world in order to create a next-generation BDMS. A first informal open source release was made four years later, in June of 2013, under the Apache Software License 2.0.

Meritocracy

The current developers are familiar with meritocratic open source development at Apache. Apache was chosen specifically because we want to encourage this style of development for the project.

Community

While AsterixDB started as a university project it has developed into a community. A number of the initial committers started contributing in academia and continue to actively participate and contribute after graduation. And we seek to further develop developer and user communities. One way to broaden the community that is ongoing is through academic collaborations (currently with IIT Mumbai in India and TU Berlin in Germany). During incubation we will also explicitly seek increased industrial participation.

Some indicators of the effort's development community and history can be found at: https://www.openhub.net/p/asterixdb/contributors?query=&sort=commits_12_mo, https://www.openhub.net/p/hyracks/contributors?query=&sort=commits_12_mo

Core Developers

The core developers of the project are diverse, although initially UC Irvine heavy (roughly 50%) due to the project's origins at UCI. The other 50% are from other academic institutions (UC Riverside and the Hebrew University in Jerusalem) and companies (Couchbase, IBM, KACST Saudi Arabia, Oracle, Saudi Aramco, X15 Software).

Alignment

Apache is, by far, the most natural home for taking the AsterixDB project forward. A large fraction of today's top Big Data technologies have their homes in Apache, including Hadoop, YARN, Pig, Hive, Spark, Flink, HBase, Cassandra and others. AsterixDB fills a significant gap – the parallel data management system gap – that exists in the Big Data open source world. It is well-aligned with a number of the Apache projects, e.g., it has strong support for accessing and indexing external data in HDFS, and it uses YARN as an answer to basic cluster resource management. AsterixDB also seeks to achieve an Apache-style development model; it is seeking a broader community of contributors and users in order to achieve its full potential and value to the Big Data community.

There are also a number of related Apache projects and dependencies that will be mentioned below in the Relationships with Other Apache products section.

Known Risks

Orphaned products

Given the current level of intellectual investment in AsterixDB, the risk of the project being abandoned is very small. The UCI/UCR faculty team leads are highly incentivized to continue development since the database groups at UC Irvine and UC Riverside are both reliant on AsterixDB as a platform for long-term graduate research projects. UC San Diego is also beginning to contribute to the code base, and a collaboration involving public health applications is forming with UCLA. The work on AsterixDB is managed via a mix of mailing list discussions supplemented by weekly project status meetings which are summarized on the mailing list. Typical (local plus Skype-in) attendance to the weekly status meetings runs at about 20 active contributors.

Inexperience with Open Source

AsterixDB and Hyracks were completely developed in Open Source under the ALv2. The source code repositories, issue tracker, and mailing lists are available on Google Code and discussions and decisions happen on the mailing lists (which is necessary due to the geographic distribution of the current developers).

Also a few of the initial committers have contributed to Apache projects. Vinayak Borkar is a committer on the Apache Helix and Apache VXQuery projects. Till Westmann is the VP VXQuery at the ASF and an IPMC member. Preston Carman and Steven Jacobs are committers on the Apache VXQuery project.

Relationships with Other Apache Products

Apache VXQuery is based on the Hyracks data-parallel runtime, which is also included in the AsterixDB code base.

AsterixDB is closely related to Apache Hadoop. Included in AsterixDB is support for accessing external data in HDFS (and Hive formats), and resource management and system administration features are in the process of being migrated to YARN.

AsterixDB's AQL query facilities offer comparable query power to Apache's Pig and Hive systems for big data analytics. AsterixDB differs in storing and indexing data and thus being able to quickly answer small and medium queries without large HDFS data scans - thereby targeting a different class of use cases.

AsterixDB's data storage and indexing facilities are similar to those of HBase, but AsterixDB differs in being a much more complete and queryable BDMS (not just a key-value style store).

AsterixDB's target use cases are not in-memory processing or iterative algorithm support, making AsterixDB complementary to the Apache Spark platform. (Spark interoperability is on our longer-term to-do wishlist.)

Homogeneous Developers

As mentioned before the current community is already organizationally and geographically distributed - and we would like to increase the heterogeneity.

Reliance on Salaried Developers

Of the initial committers only 3 are full-time UCI staff. The other committers are a mix of students, alumni who continue to contribute to the effort, and individuals working with permission part-time (or in spare time) on this project.

A Excessive Fascination with the Apache Brand

We believe in the processes, systems, and framework Apache has put in place. Apache is also known to foster a great community around their projects and provide exposure. While brand is important, our fascination with it is not excessive. We believe that the ASF is the right home for AsterixDB and that having AsterixDB inside of the ASF will lead to a better long-term outcome for the Big Data community.

Documentation

Documentation and publications related to AsterixDB can be found at http://asterixdb.ics.uci.edu/.

Initial Source

Current source resides in Google code: https://code.google.com/p/asterixdb/ (query language and upper system layers) and https://code.google.com/p/hyracks/ (dataflow runtime system and storage management libraries).

External Dependencies

AsterixDB depends on a number of Apache projects:

- Ant
- Avro
- ApacheDB JDO
- Commons
- Derby
- Hadoop
- Hive
- HTTPComponents
- Jakarta ORO
- Maven
- Tomcat
- Thrift
- Velocity
- Wicket
- Xerces

and other open source projects (organized by license):

- ALv2:
    - Jackson
    - Google Guava
    - Google Guice
    - JSON-simple
    - BoneCP
    - Microsoft Azure SDK
    - Netty
    - Rome
    - !JetS3t
    - Groovy
    - Jettison
    - Plexus
    - Datanucleus (JDO)
    - Jetty
    - Twitter4J
    - Snappy-java
- BSD:
    - Antlr
    - ObjectWeb ASM
    - Protobuf
    - JSCH
    - JavaCC
    - Paranamer
    - JLine
    - Stax
    - StringTemplate
    - xmlEnc
- MIT
    - AppAssembler
    - SimpleLog4J
- CDDL 1.0
    - Java Activation Framework
    - Java Transactions
    - Java Servlet API

    - Grizzly
    - gmbal
    - Glassfish
- CDDL 1.1
    - Jersey
    - JAXB Reference Implementation
- JSON License
    - JSON
- EPL 1.0
    - JUnit
- JDOM License
    - JDOM
- Public Domain
    - xz
    - AOPAlliance

As all dependencies are managed using Apache Maven, none of the external libraries need to be packaged in a source distribution.

Required Resources

Developer and user mailing lists

- private@asterixdb.incubator.apache.org (with moderated subscriptions)
- commits@asterixdb.incubator.apache.org
- dev@asterixdb.incubator.apache.org
- users@asterixdb.incubator.apache.org

A git repository

https://git-wip-us.apache.org/repos/asf/incubator-asterixdb.git

A JIRA issue tracker

https://issues.apache.org/jira/browse/ASTERIXDB

Initial Committers

The following is a list of the planned initial Apache committers (the active subset of the committers for the current repository at Google code).

- Abdullah Alamoudi (bamousaa@gmail.com)
- Cameron Samak (eufery@gmail.com)
- Chen Li (chenli@gmail.com)
- Ian Maxon (imaxon@uci.edu)
- Inci Cetindil (icetindil@gmail.com)
- Ildar Absalyamov (ildar.absalyamov@gmail.com)
- Jianfeng Jia (jianfeng.jia@gmail.com)
- Keren Ouaknine (kereno@gmail.com)
- Markus Dreseler (apache@dreseler.de)
- Mike Carey (dtabass@apache.org)
- Murtadha Hubail (hubailmor@gmail.com)
- Pouria Pirzadeh (pouria.pirzadeh@gmail.com)
- Preston Carman (prestonc@apache.org)
- Raman Grover (ramangrover29@gmail.com)
- Sattam Alsubaiee (salsubaiee@gmail.com)
- Steven Jacobs (sjaco002@apache.org)
- Taewoo Kim (wangsaeu@gmail.com)
- Till Westmann (tillw@apache.org)
- Vassilis Tsotras (tsotras@cs.ucr.edu)
- Vinayak Borkar (vinayakb@apache.org)
- Yingyi Bu (buyingyi@gmail.com)
- Young-Seok Kim (kisskys@gmail.com)
- Zach Heilbron (zheilbron@gmail.com)

Affiliations

UC Irvine

- Mike Carey
- Chen Li
- Ian Maxon
- Inci Cetindil
- Yingyi Bu
- Raman Grover

- Pouria Pirzadeh
- Young-Seok Kim
- Cameron Samak
- Taewoo Kim
- Jianfeng Jia
- Murtadha Hubail
- Markus Dreseler

UC Riverside

- Ildar Absalyamov
- Preston Carman
- Steven Jacobs
- Vassilis Tsotras

Hebrew University

- Keren Ouaknine

Oracle

- Till Westmann

X15 Software

- Vinayak Borkar
- Zach Heilbron

KACST Saudi Arabia

- Sattam Alsubaiee

Saudi Aramco

- Abdullah Alamoudi

Carey, Li, and Maxon are full-time UCI (UC Irvine) staff, Tsotras is full-time UCR (UC Riverside) staff, with the remaining UCI and UCR affiliates being students. The non-UC committers are a mix of alumni who continue to contribute to the effort and individuals working with permission part-time (or in spare time) on this project.

Sponsors

Champion

Chris Mattmann (NASA/JPL)

Nominated Mentors

- Henry Saputra
- Jochen Wiedmann
- Ted Dunning
- Ate Douma

Sponsoring Entity

The Apache Incubator