Benefits of CEDA

David Barrett-Lennard
Cedanet Pty Ltd
Perth, Western Australia
david.barrettlennard@cedanet.com.au

May 5, 2015

1 What is CEDA?

CEDA is a high performance database technology suitable for the management of all sorts of data, such as

- relational data
- spatial data for mining software
- CAD drawings
- word processing documents.
- multi-resolution terapixel images

The perfectly smooth panning and zooming of images on the scale of *Google Earth* on a low performance laptop highlights the exceptional performance characteristics of the database engine.

CEDA can scale from one user to many thousands of users which interactively edit very complex and large amounts of data in applications that feel as responsive as single user applications. The data is replicated and synchronised efficiently, and users are able to view and edit their local copy of the data independently of network latency and network failures. This is achieved using *Operational Transformation*. The unique and revolutionary algorithms in CEDA typically merge hours of off-line data entry in a fraction of a second.

2 Replication

Data is *replicated*, meaning that each computer has its own independent copy of the data. The data is *synchronised* through the exchange of *operations* (also called deltas) on the data which are sent over the network. The operations can be very fine-grained, meaning for example that operations record changes down to the level of insertions or deletions of individual characters in a text field. Therefore when users interactively edit some shared data they get to see each others edits in real time, such as typing in a text field or manipulating objects with the mouse. Network bandwidth utilisation is excellent because synchronisation involves deltas which only record the changes, irrespective of the total amount of replicated data.

Operations on the same data can be generated at different sites regardless of network partitions. In the database theory literature this is called *multi-master replication* and is known to be highly desirable but very challenging to implement. Indeed there have been thousands of papers on the subject in the past 40 years. It is like the holy grail for data management systems, especially large distributed systems.

It is also called *update-anywhere-anytime* replication, because a user is always able to update their local copy of the data, even if they are disconnected from other computers. Indeed the network can be very unreliable, dropping out intermittently all the time, and yet users continue working on their local copy, immune to these problems. The editing experience is unaffected by network latency or disconnections. It means multi-user software is just as responsive as single user software.

Multi-master replication is known to be difficult because of some negative results which have been established in the literature, such as the *CAP theorem* which shows that it is impossible for a distributed database system to guarantee all three of the following

- consistency
- availability
- partition tolerance.

In other words any system must pick two and give up on the third. CEDA addresses the limitations of the CAP theorem by allowing sites to temporarily diverge as operations are performed in different orders. This is sometimes called *eventual consistency*. Once all sites have received all operations they necessarily converge to the same state. CEDA does not compromise on availability and partition tolerance (in contrast to systems which do are therefore are fragile). When there is a network failure users are able to continue updating their local copies of the data, they are *autonomous*. The algorithms are very robust, and allow for redundant data paths, database rollback, changes to the network topology, failed connections and numerous other failure conditions.

In fact CEDA is well suited to replication in extremely unreliable networks. It even allows connections to be broken every few seconds and yet allows robust synchronisation of replicated data. This has been proven to work with reconnections in arbitrary network topologies that change over time. Computers can even connect that have never directly connected before in the past and exchange operations that were received from other peers. The CEDA replication system was first implemented 8 years ago and has been subjected to ongoing, heavy testing with billions of randomly generated operations on randomly generated network topologies with randomly generated reconnections.

Another negative result in the literature is a paper showing that under quite general assumptions replication is not scalable be-

cause the computational load increases 1000 fold when the system is 10 times larger. This can easily mean a system cannot keep up with a very modest transaction rate, much to the surprise of its developers. Such a situation is unrecoverable because the load increases as the divergence in the copies of the database increases. As a result many systems tend to only use master-slave replication. This means updates can only be applied to one computer (the "master") and updates only propagate in one direction to all the "slaves". This is quite limiting compared to update-anywhere-anytime replication. E.g. users cannot work if they cannot connect to the master and the data entry application may seem sluggish because of network latency (i.e. the time for messages to be sent to and from the master over the network).

Nevertheless CEDA has a computational load which is *proportional* to the size of the system, possible because it avoids the assumptions in the literature than imply replication cannot scale. In fact the algorithms are extraordinarily efficient. A single server can merge millions of operations per second received from 50000 peers. Data ingestion rate is measured in hundreds of megabytes per second on a typical server. In the database community it is commonly thought this isn't possible.

Google have tried to support update-anywhere-anytime with Google Wave, a project that caught the interest of industry experts for its exciting proposal to use Operational Transformation to achieve multi-master replication, but their solution doesnt satisfy a mathematical property in the literature called *TP2*, which means it is not able to merge changes in arbitrary orders for arbitrary network topologies. There is no doubt that Google are using inferior algorithms to CEDA for distributed computing, and this is even though they have developed their solution years after the CEDA implementation was developed and fully tested.

CEDA was compared to ObjectStore (a popular object oriented DBMS) in 2004 and CEDA was found to achieve 100 times the transaction throughput in a multi-user database system on a LAN. The benefits of CEDA would have been even greater on a WAN. This is essentially because CEDA uses multi-master replication with fully asynchronous operations, whereas ObjectStore uses distributed transactions, multi-phase commit and pessimistic locking. ObjectStore is using the conventional approach still emphasised in the database literature today, but which exhibits both poor performance and can't be made robust to network partitions because of the theoretical impossibility of guaranteeing all sites agree to commit a distributed transaction or not when the network is unreliable.

3 Development Environment

Software developers using the CEDA technology only need to define the data schema and the CEDA DBMS takes care of the rest:

- persistence in a local database
- caching of data in memory and LRU eviction
- bindings to languages like C++ and Python
- replication and synchronisation
- real time collaborative editing of the data
- working off-line and merging changes when coming on-line
- branching and merging

Developers using CEDA don't have to be concerned with conflicts when multiple users edit the same data, or what happens when connections are broken, or computers roll back to earlier states after crash recovery, or the myriad other problems that normally make developers avoid multi-master data replication despite its high desirability.

4 Log structured Store

CEDA uses a *Log Structured Store* to persist data on secondary storage. It achieves read/write performance unmatched by other database technologies. It has been found to outperform BTrieve by a significant factor and yet Btrieve is supposed to be one of the fastest database systems in the world.

Conventional databases use something called *Write Ahead Logging* (WAL) to allow a server process to fail at any time and yet the database is able to recover back to a consistent state when the server is restarted. For that reason it is actually unsafe for products like SQL Server or Oracle to use stock hardware without disabling the hard-disk write caches which can reorder writes (but that is rarely done because write performance would drop by a factor of 10 or more - maybe even 100). By contrast CEDA uses a far more resilient and efficient system for crash recovery that is almost independent of the order of writes and therefore allows for crash recovery to be performed without needing to disable the write cache on the hard-disk. The crash recovery system used by CEDA has been executed billions of times in carefully written unit tests to ensure it is bug free. It is this attention to detail than has allowed the database technology to have a zero bug report count despite nearly a decade of commercial use.

The company RungePinkcockMinarco looked at using a number of database technologies (such as Microsoft SQL Server and Oracle) for storing block models of mine sites before choosing BTrieve which gave superior performance. Indeed BTrieve is marketed as a lightning-fast transactional interface that uses a MicroKernel to achieve high performance through features such as internal indexing algorithms that cache pages for fast data retrieval and updates, and automatic index balancing to maintain fast data access.

In 2005 an engineer at RungePincockMinarco (Ian Mega) compared CEDA and BTrieve to determine whether they should convert their already mature XPAC (versions 7.7 and 8.0) product over to using CEDA. Their website is at `http://www.rpmglobal.com/`.

Ian Mega stated the following in an internal email to his project manager Simon Cleary:

> *Got some timing on LSS and Btrieve (BT) for 7.7 & 8.0. Ive had some help from Fractal and the summary is that LSS is performing very, very, very well.*
>
> *You ask, is there a LSS cache? Well indeed there is. You have to set it to a fixed size (we would probably give this option to the user) when opening the database. I have only tried this in 8.0 and the results are very exceptional for LSS*
>
> *Putting LSS in 8.0 is a no brainer even though it will take a month or so of work.*

This was after he found that it took over 2 minutes to write a block model in Btrieve and just 6 seconds using the CEDA LSS. He also found read performance was generally up to 7 times faster using CEDA.

CEDA is remarkable as well in that there hasn't been a single bug found in the CEDA libraries used for the XPAC, XACT, MKP, BLOCK AGG and OPMS mining software products, since it was first released in 2005. That is unusual for a sophisticated database technology.

In the past three years Rio Tinto (the 4th largest mining company in the world and the 153rd largest company in general) have been using CEDA to store large amounts of data (multiple terabytes). CEDA scales very well to extremely large databases and allows for thousands of terabytes - in fact read/write performance is essentially independent of the database size. This has allowed Rio Tinto to visualise cross sections in very large block models in real time, something which is simply not possible with other commercially available database management systems.