Retrieving a Sorted List of Hundreds Closest Relatives from FamilySearch Family Tree in Seconds

Ben Baker
FamilySearch
bakerb@familysearch.org

ABSTRACT

Whether a family history user is beginning to build their family tree or working in a well-developed tree already containing thousands of relatives, retaining context of how all of their relatives are related to them is often difficult. To aid users with this problem, many genealogy products include relationship calculators to show the relationships between two persons.

However, these relationship calculators typically suffer from one or more of the following deficiencies:

1. A relationship calculation must be initiated between two persons in an ad hoc manner and repeated for a different set of two persons
2. It is not possible to sort a list of arbitrarily related persons by closeness
3. Relationship calculations through marriages are not possible
4. Data to perform the relationship calculations must be pre-calculated and stored to be performant enough for on-demand calculation in an enterprise system

In previous work introducing a metric called Weighted Relationship Distance (WRD), the author presented a method to address the above deficiencies with the exception of still requiring data to be stored locally to retrieve results in a practical amount of time.

This paper demonstrates the possibility of leveraging the scalability of NoSQL databases to address the final deficiency of current relationship calculators. A prototype implementation has been developed using an Apache Cassandra based implementation of data in FamilySearch Family Tree. This prototype has been shown to be effective at retrieving a sorted list of hundreds of the closest relatives of a particular person in seconds. Such scalable performance will enable many applications to help users better understand and more effectively work with their own relatives.

1. INTRODUCTION

1.1 Previous Work

In previous work by the author, a *Weighted Relationship Distance (WRD)* metric was introduced as a unified method to compute relationship distance for related persons connected through direct, collateral and spousal lines based on closeness of relation. [1]

This metric is defined as a function of three distances (g, c, m) from a base person in a connected family tree graph, each with an associated weighting factor (α, β, γ). These distances and weighting factors are defined as follows:

g – Generational or "vertical" distance
  Number of generations from base person
c – Collateral or "horizontal" distance
  Minimum generations to a closest common ancestor
m – Marriage distance
  Number of marriages between base person
α, β, γ – Weighting factors to control growth rates

These distance and weighting factors are applied in Equation (1) below to compute the weighted relationship distance between two persons.

$$WRD(g,c,m) = \alpha(|g| + 1)e^{\beta c}e^{\gamma m} \quad (1)$$

1.2 Relationship Calculator Deficiencies

Most relationship calculators function by having a user specify two persons in a family tree and determining their relationship(s). While this helps a user to understand the context of how two persons are related, often using the user as one of the persons, it fails to answer broader questions such as "who are all of the 1st cousins of person X?" or "How many great uncles do I have?"

Similarly, even if a user had a list of persons most closely related to them, it is not possible to sort persons with different relationships by closeness without a metric to determine who is more closely related. For example, which is a closer genealogical relationship, a great aunt or a great grandfather? Providing sorting capabilities would enhance a user's ability to focus on relatives in priority order, whether for suggesting areas to work on or notifying users of changes in their tree they may be interested in.

An additional deficiency of many relationship calculators is that they typically only perform one type of relationship calculation. Specifically, they find the closest common ancestor between the two persons and show the cousin relationship between them. This is often used to find distant relationships to prominent persons. However, only supporting this type of calculation misses obvious relatively close relationships not related by blood such as a father-in-law or second wife of a 3rd great grandfather.

Computing the weighted relationship distance metric across a set of persons related to a particular person addresses all three of these deficiencies. Specifically, instead of asking how two people are

related, one may ask the question "who are all of the closest relatives of a person up to a specific distance threshold?" The result set of this operation represents the closest relatives of a person.

Because the weighted relationship distance metric produces a single value for any relationship between two persons, a list of related persons may be sorted by relative closeness. This enables many applications including creating a to-do list for a user starting with closest relatives first, notification of areas a user may be interested in and much more.

An additional feature of the weighted relationship distance metric is that relationship calculations through marriages are possible. For example, a typical relationship calculator when presented with a base person and his father-in-law would attempt to find the closest common ancestor between these persons instead of the closer in-law relationship.

However, while showing great promise for many applications, the primary deficiency exhibited by the initial prototype using weighted relationship distance calculations is that in order to retrieve a substantial number of relatives, a local RootsMagic database was being used.

1.3 FamilySearch Family Tree

FamilySearch Family Tree contains nearly 1 billion persons and previous research found the largest connected component to contain tens of millions of persons. [2] Despite this, only a small fraction of these persons, perhaps thousands or tens of thousands are related to a particular user in a degree of closeness where the relationship is relevant. Family Tree users have repeatedly requested relationship calculation features, including an ability to search their portion of the tree for various items.

As an employee of FamilySearch, the author desired to show this relationship calculation method in conjunction with FamilySearch Family Tree. Initial attempts to utilize the current FamilySearch Family Tree APIs for this purpose found execution times to be in minutes and the load on the system quite high due to the high number of person and relationship reads.

FamilySearch does support limited relationship calculation features, however. These features are available in the Memories and Temple Opportunities portions on familysearch.org and are powered by a service called the scope of interest (SOI). This service pre-calculates a user's direct ancestors up four generations and each of their descendants down two generations.

The main reasons these relationship calculation features are not available in other FamilySearch products, including Family Tree, is because of the limited nature of the area available for calculation and the performance and stability of the scope of interest service.

Due to the limitations of the current Family Tree ecosystem, the author realized a more scalable and performant solution was needed to be able to effectively work with an enterprise web-based family tree such as FamilySearch Family Tree. The author previously identified a NoSQL based solution as promising to enable on-demand relationship distance calculations involving hundreds or thousands or persons. [1]

1.4 Apache Cassandra-Based Family Tree

As a separate research project at FamilySearch, a group has been re-examining the architectural stack for FamilySearch Family Tree. The internal code name for this research project is Eureka. This team is working to provide a solution that will greatly improve the scale, performance and agility of FamilySearch Family Tree to better address FamilySearch users' needs.

Specifically, the Eureka team has been investigating replacing the Oracle relational database management system currently used by Family Tree with a NoSQL database management system. Initial prototypes show great promise to significantly improve the scalability and performance of Family Tree. This research team initially produced a prototype using MongoDB [3], but Apache Cassandra is now the database of choice for this research project. [4]

The increased performance and scalability of the Cassandra based system enables new features not previously possible in the Oracle based system. Besides increased scalability from horizontal scaling, the denormalization of data also results in fewer reads due to a person call returning aggregate data, including the one-hop relatives of a person.

This paper examines a prototype relatives API running on this platform to show the potential of retrieving hundreds of closest relatives of a person in Family Tree in seconds.

2. PROPOSED SYSTEM

The prototype relatives API has been added to the current Cassandra based test system utilizing FamilySearch Family Tree data. This system consists of a 5-node Cassandra cluster and contains the same nearly 1 billion persons found in FamilySearch Family Tree.

2.1 Available Methods

A new RESTful web service endpoint has been added to the Eureka system, with three methods for retrieving relatives of persons. The focus of this paper is on the first method, but the other two will be implemented in the future to provide powerful relationship calculation abilities for users of FamilySearch Family Tree.

The following method is currently available on this endpoint.

Method / Description
GET relatives/{id}
Retrieve the closest relatives to a person up to a specified maximum weighted relationship distance.

Parameters
double maxDistance – Optional query parameter to specify the maximum weighted relationship distance to return relatives of the person up to.

Returns
A list of RelativePerson objects, sorted by increasing distance

Additional methods on this endpoint are planned to get the closest common ancestor between any two persons and to get the closest relation between two persons, returning all of the persons along the path along with weighted relationship distance information for all persons.

2.2 RelativePerson Object

The results from a relatives endpoint call is a list of RelativePerson objects containing basic data about each person, enough data to sort relatives and understand how they are all related. Below is an example RelativePerson object in JSON format.


```
{
    "name": "George Dean Cockle",
    "id": "KWJX-VMC",
    "lifespan": "1901 - 1970",
    "fatherIds": [
      "K2V7-PV5"
    ],
    "motherIds": [
      "K2V7-P92"
    ],
    "spouseIds": [
      "MM8P-MGS",
      "KWBB-7G9"
    ],
    "familyName": "Cockle",
    "givenName": "George Dean",
    "gender": "MALE",
    "childIds": [

    ],
    "relative": {
      "id": "K2V7-PV5",
      "name": "John Cockle",
      "lifespan": "1852 - 1921",
      "relativeRole": "CHILD"
    },
    "weightedRelationshipDistance": {
      "starRanking": 8,
      "simpleRelationshipDistance": "3",
      "weightedRelationshipDistance": "8.155",
      "generationDistance": 2,
      "collateralDistance": 1,
      "marriageDistance": "0"
    },
    "relativeDescription": "Great Uncle"
}
```


The relative portion of the RelativePerson object contains the relative from which the person's relation was discovered. For example, in the sample object above, George Dean Cockle is a child of John Cockle (PID K2V7-PV5). These relations can be used to draw relationship paths from one person to another as well as supporting a feature to display all one-hop relatives of a person like the Related People feature currently in FamilySearch Memories.

The weightedRelationshipDistance portion of the RelativePerson object contains information about how close the person is related to the base person. The starRanking simplifies weighted relationship distance values to a scale of 0-10 where 10 is a very close relation and 0 a distant relation. Simple relationship distance is the sum of the absolute value of the three distances (generation, collateral and marriage) and the weightedRelationshipDistance is the value computed according to Equation (1).

Finally, the relative description is a natural text description of how the person is related to the base person. These descriptions can relatively easily be determined from the gender, distances and relatives of the person. The author expects these descriptions to be used throughout familysearch.org in the future to help users better understand their relation to persons in the tree.

3. RESULTS AND APPLICATIONS

3.1 Initial Performance Results

The initial prototype supporting retrieving hundreds of closest relatives performs much better than any other relationship calculation service in use at FamilySearch. However, performance tuning is still necessary to make this service more useful for production use. The current performance results on the author's pedigree on the Eureka research system are:

Maximum WRD	Num Persons	Mean Execution Time
5.0	43	1.95s
10.0	167	22.1s
15.0	554	89.8s

For comparison purposes, the average time to retrieve data for a pedigree containing 10-15 persons in FamilySearch Family Tree is typically about 3-5 seconds. The Eureka team has show sub-second pedigree load time involving the same number of persons and loading up to 12 generations in several seconds.

3.2 Applications Enabled

The prototype system described in this paper begins to demonstrate capabilities that enable new applications not yet seen in FamilySearch Family Tree or other online enterprise family tree systems. Some examples of applications enabled by the method include:

- Identifying the closest relatives where historical record hints have been identified but not attached yet as sources.
- Promoting e-mail campaigns to point out what others have added to your relatives such as new photos, sources, stories, etc. to draw users back to the site.
- Easily identifying end of line relatives and likely places for successful descendancy research.
- Facilitate LDS temple work for closest relatives first and sharing more distantly related people with others.
- Producing to-do lists sorted by closeness of relation on any task a user may want to undertake (Ex. fixing data anomalies, merging possible duplicates, providing missing data, etc.)
- Automatic watching of relatives via e-mail alerts based on closeness.

- Applications across a set of users (Ex. closeness of relation to the user for all LDS temple submissions or photo uploads on FamilySearch)
- Sorting items such as memories, watched persons, LDS temple reservation lists, etc. in order of those closest to the user.
- Pointing out relatives who have been identified as prominent or have participated in significant events.

The current expectation is that a closest relatives call will be made when a user logs into the system, perhaps running in the background if necessary. A mechanism to refresh this set of relatives by making another call should also be provided. The resulting relatives list would be stored in the browser and/or as an object associated to the user. Doing this would enable even better performance because the list could be used for many applications without making another call to the web service and any data necessary for relationship calculations within the set is already available.

4. CONCLUSIONS AND FUTURE WORK

This work has shown a more performant method for retrieving hundreds of the closest relatives of a person in an enterprise web system than developed in previous work. This method is also much more powerful than two person relationship calculations typical to most genealogy software. It is expected that additional refining of the graph traversal algorithm will yield even better performance, enabling many powerful applications within FamilySearch Family Tree.

As FamilySearch transitions Family Tree from the current Oracle based system to an Apache Cassandra based system, it is expected that a one-way synchronization from the Oracle based system to the Cassandra based system will be used to keep data current in the new system as features are stabilized. Doing this enables read-only operations on Family Tree data from the Apache Cassandra based system, including the relative calculations described in this paper. The author intends to petition FamilySearch to utilize the work described in this paper before Family Tree is completely moved to the Apache Cassandra based system, enabling patrons to more effectively work with their own relatives.

5. REFERENCES

[1] *Beyond the Relationship Calculator – Using a Weighted Relationship Distance Metric to Prioritize, Categorize and Visualize Relatives.* **Ben Baker.** 2013 Family History Technology Workshop. http://fht.byu.edu/prev_workshops/workshop13/papers/baker-beyond-fhtw2013.pdf [Accessed Mar 2014]

[2] *Analyzing the Family Tree.* **Daniel W. Rapp and Michael P. Jones.** 2012 Family History Technology Workshop. http://fht.byu.edu/prev_workshops/workshop12/papers/5.2%20AnalyzingTheFamilyTree.pdf [Accessed Mar 2014]

[3] *HumMONGOus Family Tree Application Running on a MongoDB Cluster – A Case Study.* **Randy Bliss, Judson Flamm, Randall Johnson, Tom Valletta.** SORT Conference 2012

[4] *Increasing Scale and Performance Through Data Denormalization.* **Arn Perkins, James Gates, Jason Daniels, John Sumsion, Randy Bliss.** SORT Conference 2013.