Project List

Vassilios Lekakis
lex@cs

1 The List

1. Consistency measurement study for user-centric cloud storage applications.
2. Energy constrained consistency.
3. T.Rex in the wild.
4. Distributed collaboration.

2 Consistency measurement study

In this project I'm interested to find out the performance users get out of the cloud storage application. The major questions I will try to answer are:

1. What's the probability of users to see stale data.
2. What type of eventual consistency is supported from these applications.
3. Examine the tussle between cloud and distributed based solutions.
4. How the performance of a distributed system compared to the current performance of cloud based solutions.

2.1 Plan of action

In order to achieve the goal of this project I need to complete the following steps.

1. Data collecting phase:
   - Experiment with a single pair of a reader/writer. Changing networking locations
   - Experiment with a single pair of a reader/writer with different write rates; write rate is the number of writes per minute.
   - Experiment with multiple readers/single writer.
   - Experiment with a single reader but many writers.
2. Data Analysis:
   - Learn about the probability of stale data on the different systems.
   - Are these systems affected from network location, data size, update rate?
   - How these systems handle conflicts?
   - What type of eventual consistency they support? For example which one of the following consistency schemes *read-your-writes*, *session-consistency*, *monotonic-read*.

3. Comparison with a distributed system:
    - Find out if we can tune T.Rex (or T.Rex without security) to achieve the same performance or even better.
    - Find out the impact of the topologies on T.Rex and see how different update rates affect the performance of a distributed storage system. This last step is not novel at all, but will make the study complete.
4. Lessons learned. Explaining the trade-offs between the two approaches.

2.2 Merit

- Learn something cool, that affects almost 200 million people.
- Test T.Rex under different topologies, update rates etc.
- Write a paper about it.

3 Energy constrained consistency

The CAP theorem states that a system cannot achieve consistency, availability and partition tolerance in the same time. Everyday we experience a variation of the CAP theorem when we are using our smartphones and tablets. Keeping all of our data consistent will probably drain the battery of the device. Enabling the networking interfaces to accept or propagate updates aggressively comes with high power consumption. Users and application developers have chosen a 'soft partitioning' to favor battery life at the expense of data consistency.

In this project I want to examine how far we can push the tussle between consistency and energy consumption, in favor of consistency. In other words, how consistent we can keep the data stored in mobile devices without exhausting their battery life. Figure 1 shows the tussle between the two straw-man approaches. The goal of this project is to achieve the strongest possible consistency scheme by minimizing the amount of energy we consume.

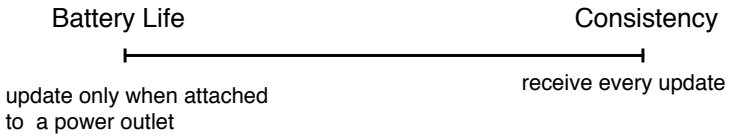

Figure 1: Tussle between power consumption and consistency. If our goal is to preserve energy the straw-man approach dictates updating only when the device is plugged into a power outlet. On the other end a straw-man solution that favors consistency is to update aggressively.

This project has similar flavor with T.Rex since I will examine how energy affects replication and consistency.

3.1 Plan of action

Hoarding: The first step is to figure out how we can do smart caching. My plan is first to test if the SEER (hoarding paper) approach works for mobile devices. If it works I have to determine if this approach is feasible. I will determine its feasibility from the power it requires to run the *observer* in a mobile device. Except the SEER approach, there are some other more modern approaches that worth examination; these

are 'Connections' (SOSP 2005-Granger) and 'Autograph' (SIGOPS Operating Systems Review-Aguilera). This study should answer the following questions:

- Is it possible to achieve accurate caching with the automatic hoarding techniques?
- Are the results as accurate as would be if the users would make manual selections?
- Is automatic hoarding a viable solution under the current energy constraints of today's devices?

Consistency Vs Update Frequency: The results of the previous phase will determine if we can use an automatic technique to hoard items in mobile devices. Ideally, at the end of the first step of the study the a mobile device will include three types of files.

- *System*: Files in this category must remain at the devices all times since are necessary for their operation.
- *Hoarded-No-Update(HNUP)*: Files that belong in this category are cached in the device but we expect that won't change frequently or at all. For example music files that the users want to have locally in their devices belong in this category. The reason to evict these files from the decive is because the hoarding system will not classify the workflow they belong as *hoardable*.
- *Hoard-Updatable(HUP)*: In this category belong all the files that need regular updating.

Based on this classification the files of interest are the ones belong in the HUP group. What we need is a smart mechanism to help us decide when to update the files of this group. A scheduling mechanism like this can be the 'Bartendr' which provides the applications with hints on when is energy efficient to sent or receive updates.

The items in the HUP group can be assigned in different consistency groups. Therefore, we can push the energy consistency trade-off even further deciding on per HUP-item basis when to perform an update. I believe this approach will favor consistency in a more targeted manner, thus, without sacrificing a lot of battery power. Maybe ideas from PBS and distributed databases will help here.

4 T.Rex in the wild

This is a simple one step project. The purpose is to make T.Rex robust enough to work for real users. The next step is to get an IRB and do a user study. The main purpose of this project is to examine how users will take advantage of a system like T.Rex. Probably, this idea doesn't really come with an intellectual merit and it will be hard to reach it to the point that real users will get T.Rex code and actually use it.

5 Distributed collaboration

Can we combine small collaborative quorums, which require strong consistency, with traditional anti-entropy based replicas that have more relaxed consistency needs? Then next step in this project is to answer if we can achieve this while eliminating information leakage. In other words how we can make T.Rex actually support these two consistency types without sacrificing security.

5.1 Plan of action

- Better understanding of consistency models. This means that the first step is to make the T.Rex consistency work and take some ideas on how expensive they are in action.
- Come up with a protocol to support interactive quorums. These might be Dynamo style quorums, when another alternative can be Paxos.

- Decide how the results of the collaboration will appear to other replicas. The question is every update that is confirmed from the nodes will be immediately updated to the rest anti-entropy nodes or the end result of the collaboration will considered a single update?
- Handle cases of concurrent updates which happen inside the strong consistency group and the rest of the eventually consistent replicas.