Terry13

Èric Casanovas

March 24, 2019

Upload your report in PDF format.
Use this LaTeX template to format the report, keeping the proposed headers.
The length of the report must not exceed **5 pages**.

1 Summary

Aquest text es tracta d'un article periodístic al diari communications of the acm. L'objectiu d'aquest article és explicar la consistencia de les dades comparant-lo amb el baseball per a que el lector ho entengui d'una forma fàcil i amena.

Aquest document ens parla sobre la relació que hi ha entre la consistencia de les dades i un partit de baseball. Comença explicant com funcionen Windows Azure Storage que té una forta consistencia de les dades, degut a que sempre ens dona l'últim valor del data object. D'altra banda trobem sistemes com Amazon Simple Storage que pensen que tenir una forta consistencia de dades provoca que sigui molt costos en termes de temps, això fa que el valor retornat sigui igual al valor de fa un temps del data object això s'anomena eventually consistent. També trobem Amazon's DynamoDB que incorpora les 2 modalitats anteriors. Doncs veiem que depenent de les nostres necessitats necessitarem que ens garanteixin una consistencia més o menys elevada. Existeixen 6 tipus:

1. Strong Consistency: Garanteix que dona el valor de l'ultima escriptura sobre l'objecte demanat. Ës la més fiable però la menys ràpida.

2. Eventual Consistency: Retorna qualsevol de les últimes escriptures sobre l'objecte demanat. És la menys fiable però la més ràpida.

3. Consistent Prefix: Retornarà un valor que probablement no sigui l'últim ja que ens pot contestar una rèplica que potser encara no ha rebut tots els writes sobre l'objecte.

4. Bounded Staleness: S'assegurarà que el resultat retornat no sigui de fa molt temps. Típicament retorna el resultat de com a màxim T minuts enrere on T el pot passar qui demana les dades.

5. Monotonic Reads: Consisteix a que si es fa una lectura d'un object les següents lectures mínim retornin la mateixa escriptura que l'ultima lectura o una posterior.

6. Read My Writes: Garanteix que la següent lectura sobre un objecte serà minim l'ultima escriptura feta pel client (o l'última feta per un altre client si ha sigut posterior).

Cadascuna de les anteriors té els seus punts forts i febles, podriem veure que en temes de consistència l'strong consistency és el millor, seguit del Bounded Staleness que ho faria també força bé, d'altra banda l'eventually consistency seria el pitjor i la resta més o menys serien consistents. Pel que fa a la performance trobem que l'strong consistency seria el pitjor degut a que trigaria molt en retornar la lectura, i el millor seria l'eventually consistency. També en temes de performance ho farien força bé el Consistent Prefix i el Monotonic Reads. Per acabar la comparació, en temes de disponibilitat de les dades tornem a trobar que l'strong consistency seria el pitjor juntament amb el Bounded Staleness i d'altra banda els que millor ho farien serien Consistent Prefix i eventually consistency.

Podem trobar un exemple amb un partit de baseball on guardarem en 2 objects (un pels runs del equip local i un pels runs del visitant), considerem que els locals marcaran primer, despres els visitants, 2 cops els locals, 1 cop els visitants i finalment 2 cops els locals. Depenent de que agafem podria ser que ens retornes resultats diferents, per exemple:

1. Strong Consistency: Retornaria únicament el resultat correcte que seria 5-2.

2. Eventual Consistency: Retornaria qualsevol dels possibles resultats amb un màxim de 2 runs pels visitants i 5 pels locals.

3. Consistent Prefix: Retornaria qualsevol dels resultats que s'han donat durant el partit (0-0, 1-0, 1-1, 2-1, 3-1, 3-2, 4-2, 5-2).

4. Bounded Staleness: Retornaria depenent del temps que fes de l'últim write del marcador.

5. Monotonic Reads: Depenent de l'ultim read que si per exemple ha sigut el 3-1, retornaria el 3-1 o qualsevol dels resultats posteriors durant el partit

6. Read My Writes: Depenent si l'ultim escriptor ha sigut qui ha demanat llegir, si es així es comportaria com el strong consistency sinó com el eventual consistency.

Ara examinarem la consistencia necessaria per cada persona involucrada en un partit de baseball:

- **Marcador:** És imprescindible que el marcador llegeixi l'última dada existent. Curiosament, encara que necessiti dades fortament consistents no precisa de lectures fortament consistents ja que com que és l'únic que modifica els valors en tindrà prou amb fer servir "read my writes". El fet de que entre diferents "run's hi hagi minuts o fins i tot hores de diferència permet que gairebé tots els altres servidors hauran rebut l'escriptura anterior i podran respondre a la següent lectura que demana el "read my writes".

- **Àrbitre:** la part important de l'àrbitre ve després que l'equip visitant hagi batejat i l'equip de casa estigui a punt de batejar. Donat que és l'última entrada (i un equip no pot puntuar negativament), l'equip de casa ja ha guanyat si està avançat al marcador. És per això que l'àrbitre necessita "strong consistency".

- **Reporter de ràdio:** no és greu que el reporter informi de resultats lleugerament erronis, però no pot donar resultats que mai hagin existit o donar un resultat "superior" i posteriorment donar-ne un inferior. És per això que utilitzariem "monotonic reads o consistent prefix".

- **Escriptor esportiu:** Com que la latència no és important en aquest cas en tenim suficient amb un "bounded staleness" de on li passarem el nombre d'hores que fa que ha acabat el partit.

- **Estadístic:** pel mateix motiu que l'escriptor, l'estadístic també en té prou amb un "bounded staleness" desde que ha acabat el partit però a l'hora de llegir estadístiques antigues que només ell ha modificat necessitarà forta consistència i ho aconseguirem amb un "read my writes".

- **Observador de dades:** donat que l'objectiu d'ús de la informació obtinguda per aquest participant no és rellevant seria suficient amb un "eventual consistency".

Al cap i a la fi tots els participants anteriors lo òptim seria "strong consistency" però per aconseguir millor rendiment i disponibilitat de les dades fem servir uns altres i d'aquesta manera el sistema no es sobrecarregarà només fent lectures de strong consistency ja que són molt costoses.
Hem vist que les sis garanties de consistència són útils cadascuna al seu cas. Diferents clients poden desitjar consistències diferents fins i tot quan accedeixen a les mateixes dades. Fins i tot les bases de dades més simples poden tenir usuaris diversos amb necessitats de consistència diferents. Els clients haurien de poder triar la consistència que desitgin.
En definitiva, el fet de només tenir "strong consistency" facilita la vida als desenvolupadors a l'hora de fer programes però reduim moltissim el rendiment del sistema i la disponibilitat de les dades i renunciar a això seria perdre massa quan un sistema és molt més gran i complicat que una simple base de dades per a guardar resultats de baseball.

2 Assessment

En la meva opinió el text ha sigut de fàcil lectura i fàcil de comprendre, tot i que una mica llarg, tot i així també es un text molt interessant i s'hauria de seguir ficant els següents cursos.