Replicated Data Consistency Explained Through Baseball

Team Members:
Ashrit Debadi
Riti Gupta
Vedashree Bhandare

Introduction

Problem to resolve

- ❏ Explore different consistency models using baseball game application.

Operating environments/ intended usage

- ❏ The score of the game is recorded in two objects in NoSQL database.
- ❏ Users - multiple participants in baseball game.

Design tradeoffs

- ❏ Tradeoff between consistency and availability and performance.
- ❏ Eventual consistency is insufficient for most of the participants, but strong consistency is neither required.

Read Consistency Guarantees

1. **Strong Consistency**
   - Read returns result of all the writes to an object.

2. **Eventual Consistency**
   - Returns any value that was written in past.
   - Weakest Consistency- returns greatest set of possible values.

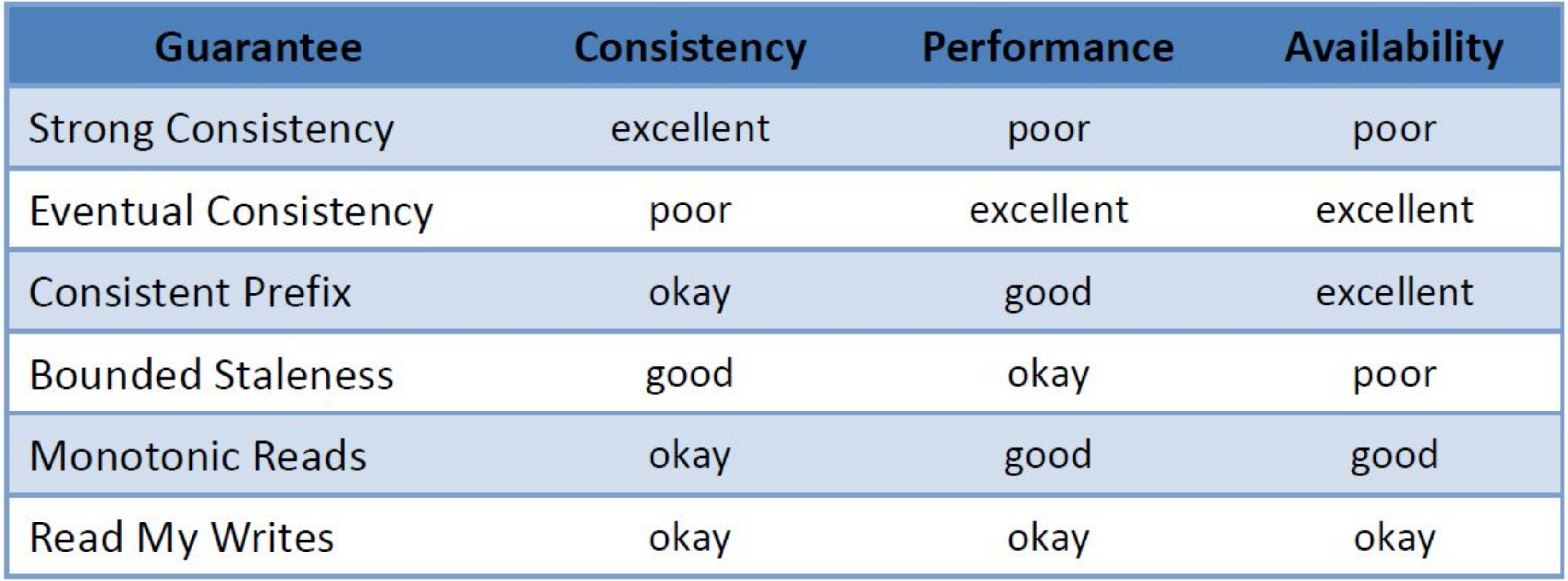
3. **Consistent Prefix**
   - Guaranteed to observe ordered sequence of writes that happened to object.
   - Similar to “snapshot isolation” consistency.

Read Consistency Guarantees

4. **Bounded Staleness**
    - Read results bounded by staleness defined by parameters like time, amount of inaccuracies, number of missing writes.

5. **Monotonic Reads**
    - Observe data values that are increasingly up to date.
    - Also referred as “Session guarantee”.

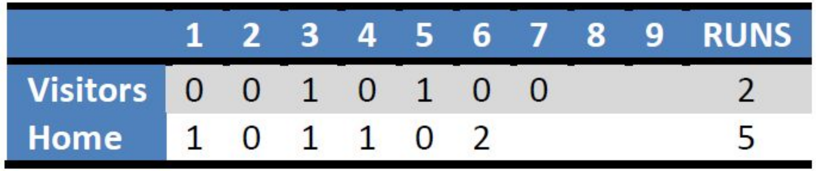
6. **Read My Writes**
    - All the writes performed by the client are visible for subsequent read operations.



Guarantee	Consistency	Performance	Availability
Strong Consistency	excellent	poor	poor
Eventual Consistency	poor	excellent	excellent
Consistent Prefix	okay	good	excellent
Bounded Staleness	good	okay	poor
Monotonic Reads	okay	good	good
Read My Writes	okay	okay	okay

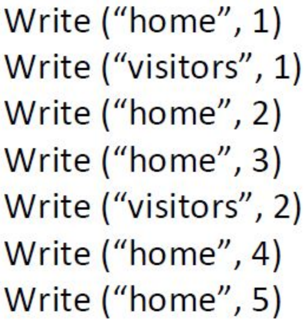
Table 2. Consistency, Performance, and Availability Trade-offs

Baseball as a Sample Application



	1	2	3	4	5	6	7	8	9	RUNS
Visitors	0	0	1	0	1	0	0			2
Home	1	0	1	1	0	2				5

Figure 3. The Line Score for this Sample Game

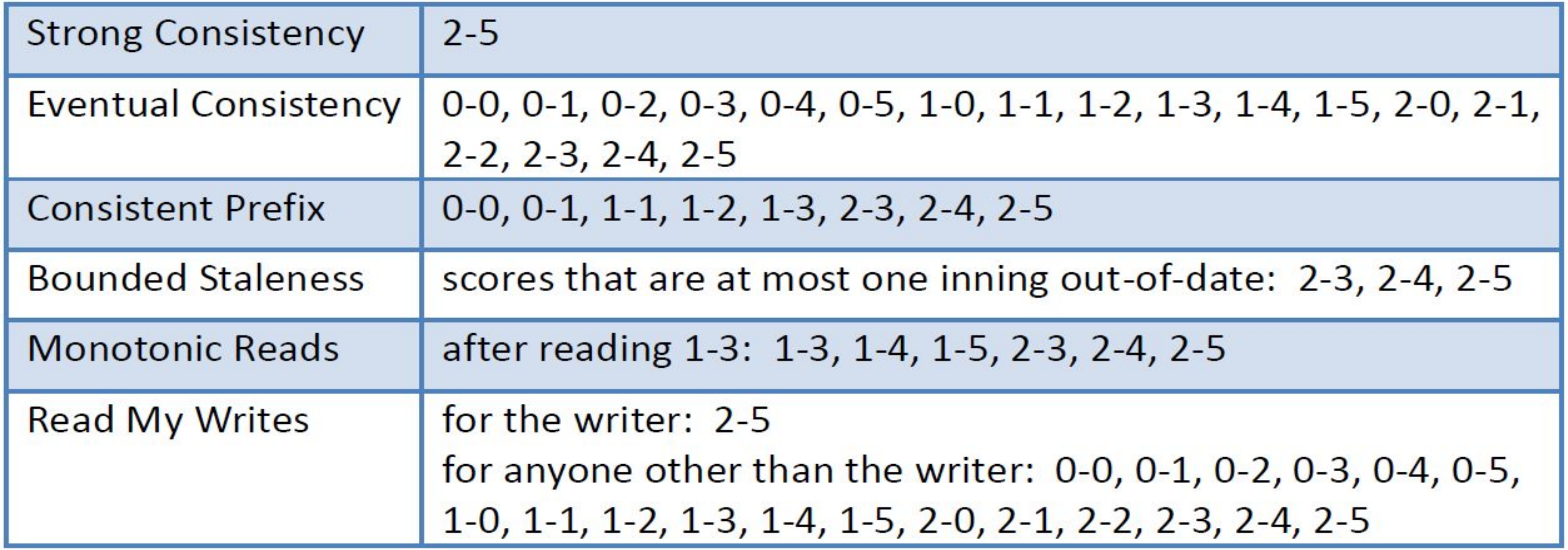


```
Write (“home”, 1)
Write (“visitors”, 1)
Write (“home”, 2)
Write (“home”, 3)
Write (“visitors”, 2)
Write (“home”, 4)
Write (“home”, 5)
```

Figure 2. Sequence of Writes for a Sample Game

- The game starts with the score of 0-0
- The visitors bat first and remain at bat until they make three outs
- Then the home team bats until it makes three outs
- This continues for nine innings

Possible Read Scores



Strong Consistency	2-5
Eventual Consistency	0-0, 0-1, 0-2, 0-3, 0-4, 0-5, 1-0, 1-1, 1-2, 1-3, 1-4, 1-5, 2-0, 2-1, 2-2, 2-3, 2-4, 2-5
Consistent Prefix	0-0, 0-1, 1-1, 1-2, 1-3, 2-3, 2-4, 2-5
Bounded Staleness	scores that are at most one inning out-of-date: 2-3, 2-4, 2-5
Monotonic Reads	after reading 1-3: 1-3, 1-4, 1-5, 2-3, 2-4, 2-5
Read My Writes	for the writer: 2-5 for anyone other than the writer: 0-0, 0-1, 0-2, 0-3, 0-4, 0-5, 1-0, 1-1, 1-2, 1-3, 1-4, 1-5, 2-0, 2-1, 2-2, 2-3, 2-4, 2-5

Table 3. Possible Scores Read for Each Consistency Guarantee

There are six different users trying to access these scores -

- ScoreKeeper,
- Umpire,
- Radio Reporter,
- Sports writer,
- Statistician
- StartWatcher accessing

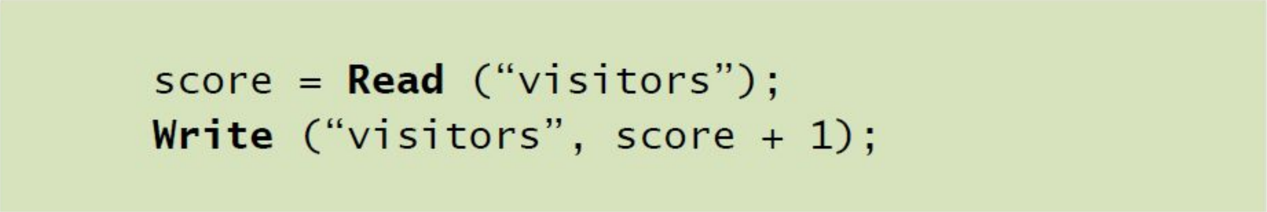
They access runs at various durations of the baseball game providing strong Consistency is not the best solution and it can be achieved by other means improving performance and Availability.

Official scorekeeper

He is responsible to update the latest score of the runs scored ,the main tasks performed by him are

- reading the last runs scored and
- writing the latest runs after scoring.

The runs that he see while reading should always be the latest values hence they should be strongly consistent but this can be achieved by Read My Writes since ScoreKeeper is the only person who is performing write operation so the system needs to find some server that has seen all the writes by the client.



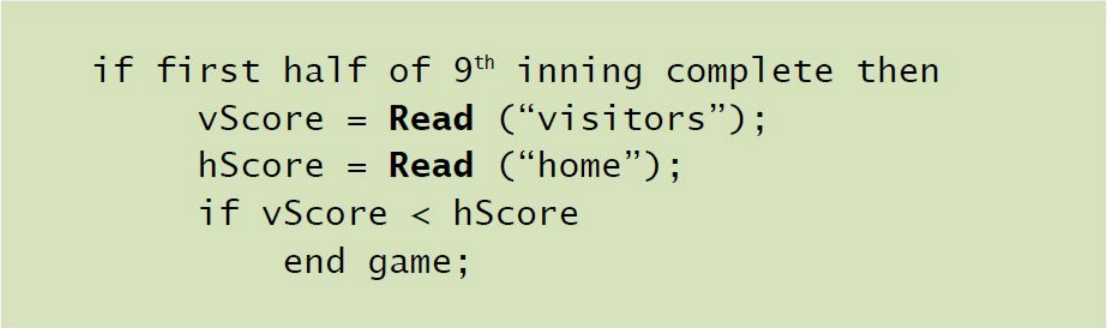
```
score = Read (“visitors”);
Write (“visitors”, score + 1);
```

Figure 4. Role of the Scorekeeper

Umpire

The main task is to check if the visitors score is less that the home score after the first half of the 9th innings so that is the visitors core is less he can end the game.

To perform this the Umpire reads the data visitors and home score and needs to have latest scores which can be achieved only by strong consistency.

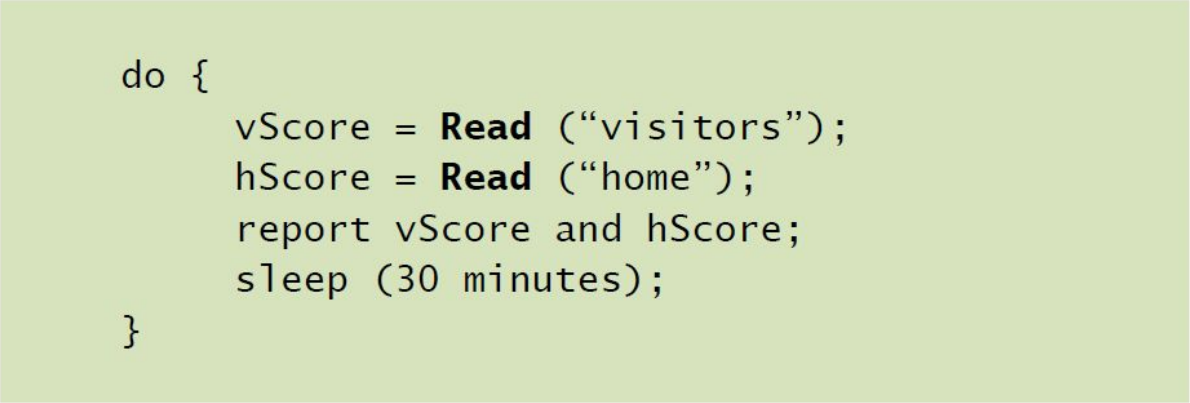


```
if first half of 9th inning complete then
    vScore = Read (“visitors”);
    hScore = Read (“home”);
    if vScore < hScore
        end game;
```

Figure 5. Role of the Umpire

Radio reporter

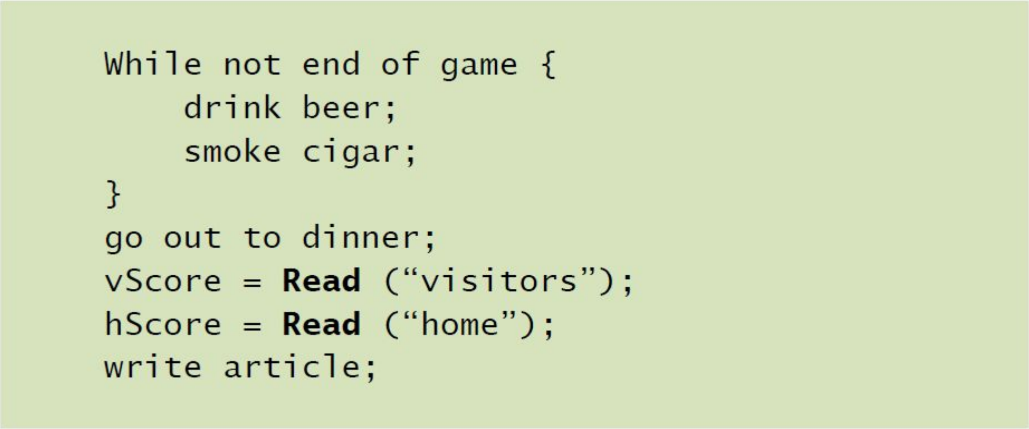
- They usually announce the score every 30 mins , performs read operation to fetch the visitors and home scores and are not required to have the latest scores but some valid score that was last updated to them.
- Some eventual consistency is required but the score has to be valid it cannot display 1-2 after showing 5-2,
- Using consistent prefix with monotonic reads where the scores which actually existed only are displayed in increasing order or
- By monotonic reads using bounded staleness with a bound time of 30 mins.



```
do {
     vScore = Read (“visitors”);
     hScore = Read (“home”);
     report vScore and hScore;
     sleep (30 minutes);
}
```

Sportswriter

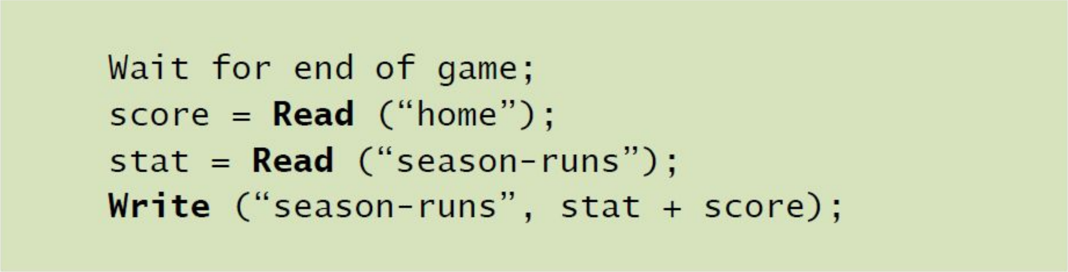
- Usually reads the score of visitors and home after the match is completed.
- He needs to get the latest scores (strong consistent score)
- By bounded staleness with a bound time of 30 mins assuming that he access the data after one hour of the match ,
- Eventually consistency might also give same results but it cannot guarantee 100% accuracy .



```
While not end of game {
    drink beer;
    smoke cigar;
}
go out to dinner;
vScore = Read (“visitors”);
hScore = Read (“home”);
write article;
```

Statistician

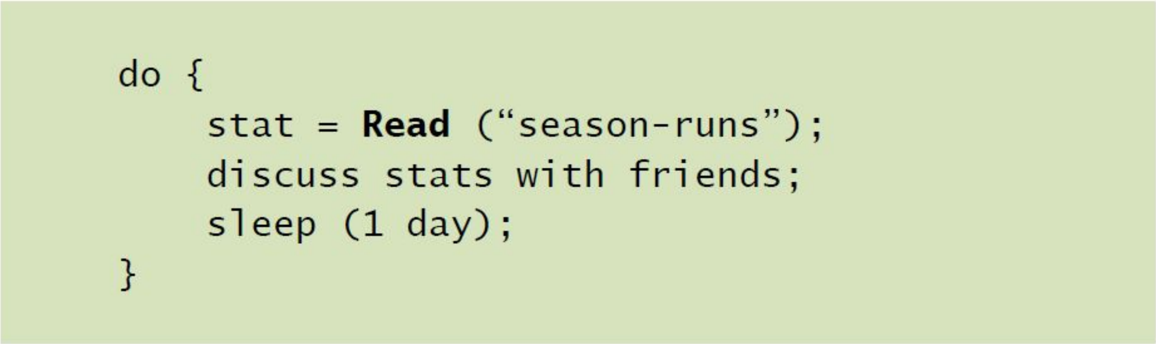
- Keeps track off all the runs scored by the team during a season
- Performs two read operations- reading the current match score of the home team , reading season score of the home team and writing the season score.
- First read she should get strong consistency read value which can be achieved by using bounded staleness read if the read is done after the match is completed.
- For the second read (season score), since she is the only person who performs write latest values can be achieved using Read My Writes .



```
Wait for end of game;
score = Read (“home”);
stat = Read (“season-runs”);
Write (“season-runs”, stat + score);
```

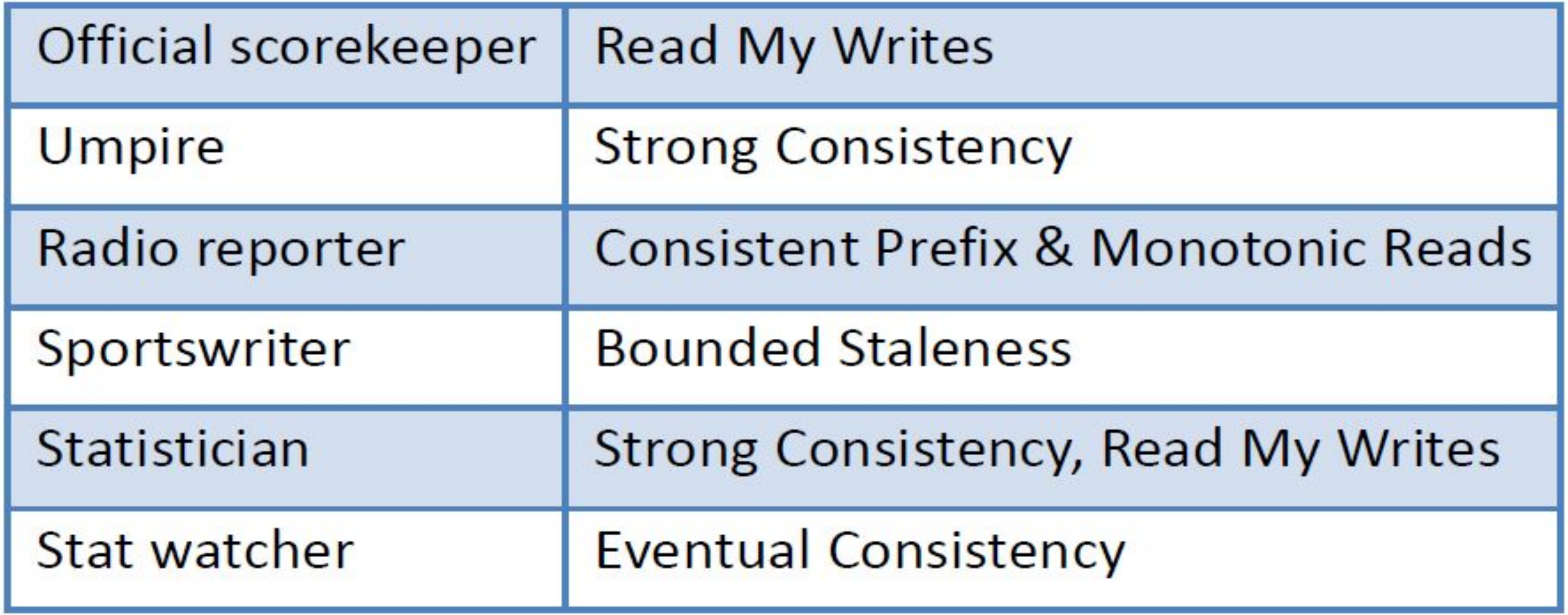
Stat watcher

- He is someone who checks the team season score say once a day by performing read on season score.
- It is okay to get old data so eventual consistency fits for this use case.



```
do {
    stat = Read (“season-runs”);
    discuss stats with friends;
    sleep (1 day);
}
```

Summary of Read Consistencies



Official scorekeeper	Read My Writes
Umpire	Strong Consistency
Radio reporter	Consistent Prefix & Monotonic Reads
Sportswriter	Bounded Staleness
Statistician	Strong Consistency, Read My Writes
Stat watcher	Eventual Consistency

CONCLUSION

- Consistency can be of various degrees :
    - Strong, Eventual, Constant Prefix, Bounded Staleness, Monotonic Reads, Read My Writes
- Different users need different consistency level
- Providing different consistency can improve performance and availability.
    - Every server need not be accessed with every write and read request
- Might be challenging for application developers to develop such a system
    - Application details need to be hidden from clients
    - Deep understanding of application is needed
    - Can be made easier by performing write operations in strict order

INSIGHT / CRITICISM

- Need of future
- Might be challenging to implement for complex applications
- Tradeoff between implementation effort and requirements should be taken into consideration