

CricBuzz

High-Level Solution Design Document

Version: 1.00

Updated: 04/09/2022

Author: Ashish Nick

1. Introduction

Cricbuzz is a cricket news website which gives information about all the games, scores etc.

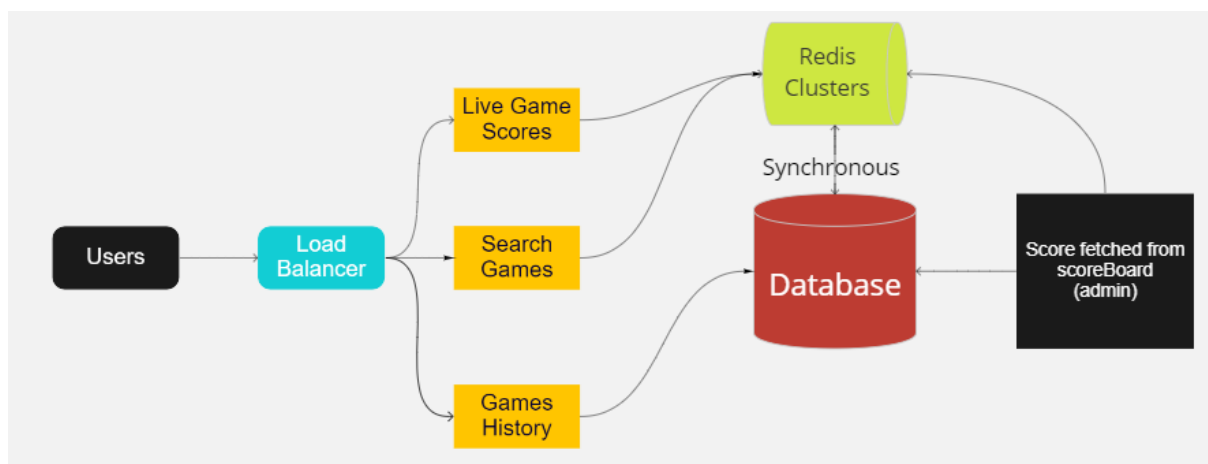
2. Requirement Summary

- Design subscription-based sports website which can display scores, game status, and history for any games.
 - User screen, live scores, live games, history
 - User should be able to search games

3. Assumptions and Prerequisites

- Cricbuzz is more used for reading purpose only. It is more read-heavy than write heavy. So let's assume:
 - Reads per sec : 600000
- We will focus more on accessing database then to write in it.

4. High-Level Design



- The interaction happens between main user portal and database and admin who updates the scores (from scoreboard directly) (as shown in figure)

4.1.Application Modules

Includes the following:

- User portal where default user can see ongoing live games
- User can see selected game live score
- User can search on the basis of games
- Server layer where all fetching and retrieving functions takes place

4.2. Transactions and User Flows

- Live games: all the games is fetched when this page shown.
 - Get all the data from REDIS as it is in sync with main database
 - REDIS is fast so we will cached trending games score on it
 - And if required then we will fetch from main database.
- Old games history: all data till yesterday are fetched.
 - Get all the data from the main database.
 - As getting history should be that fast so we will save recent (especially ongoing) games on REDIS.
- Search : only data which are related with the entered game will be fetched
 - Get all data from the REDIS if required fetch from main database.
- Load balancers: To handle the request load in peak timings, load balancers will be used.
- REDIS clusters: used to cache some data so that it can easily be available.