



Staffordshire
UNIVERSITY

Advanced Database Systems

CE00332-6

Critically discuss whether a distributed database architecture or a cloud solutions would be the best choice for Beaconside Radio

Harry Clewlow (c012952a)

Contents

1.0.0 Introduction	Pages 3
1.1.0 Introduction	
2.0.0 NoSQL Solution	Pages 3
2.1.0 NoSQL Solution Explained	
3.0.0 Distributed Database Solution	Pages 3 – 4
3.1.0 Distributed Database Solution Explained	
4.0.0 Cloud Database Solution	Pages 5
4.1.0 Cloud Database Solution Explained	
5.0.0 Recommendation to Beaconside Radio's Management	Pages 6
5.1.0 Recommendation Explained	
6.0.0 Appendix	Pages 6
6.1.0 Assumption	
6.2.0 Word Count	
7.0.0 Bibliography	Pages 6 – 7
7.1.0 Website References	

1.0.0 Introduction

1.1.0 Introduction

With the expansion of Beaconside Radio and its upcoming merger with four other local radio stations the database previously designed will be implemented for all the stations, this report will discuss where a NoSQL, distributed database or cloud solution would suit Beaconside Radio.

2.0.0 NoSQL Solution

2.1.0 NoSQL Solution Explained

A possible solution is NoSQL which allows for auto-sharding, which means that a system can be spread across several server automatically meaning the system can be very extensible, unlike a conventional distributed database which doesn't support sharding meaning a development team be needed therefore NoSQL would save on development overheads. [1]

With the radio stations needing to instantly access the database of songs, NoSQL allows for the data and query loads to be balanced across the servers automatically because of this when a server goes offline it can be replaced without any disruption to the users. [1]

Although NoSQL has many advantages I wouldn't recommend it for Beaconside Radio, mainly due to unstructured data models which wouldn't be practical as you would want a standard format for all the radio stations to use. Also the fact that there is no standard interface for a NoSQL database this would incur extra development overheads.

3.0.0 Distributed Database Solution

3.1.0 Distributed Database Solution Explained

Distributed databases have several different architectures such as Homogenous Distributed Database Systems where all systems have the same software and cooperate with each other, Heterogeneous Distributed Database Systems where the system may have different hardware and software or data structures this leads to some systems being incompatible with each other. [4]

I would recommend a Heterogeneous Distributed database as being the most appropriate architecture for Beaconside Radio's system, this would allow each system to simultaneously access and modify data in a distributed environment which will allow each station to access all the songs in Beaconside Radio's database.

A Distributed database offers many benefits such as organisational structure where you can have a uniform format across the whole of the system, allowing for local or global inquiries over a number of different systems (radio stations). With Beaconside Radio expanding distributed databases will allow for modular growth where a new station can be added without affecting other stations and gives the system great extensibility, but with this type of system the cost of expansion is new hardware when required to deal with the increase in storage and processing on the centralized database. There is also improved share ability and local autonomy where each station can access the other stations files if station 1 wants to play a song that station has they can access that song and play it with ease or vice versa, with this if one of the systems at station 1 fails the other stations are not affected directly they just can't access the songs for station 1 but can keep playing the songs on their system. [5]

The diagram shown below represents an example of a Heterogeneous distributed database where the replication of songs is handled by all the databases interacting with each other allow the system to be extended easily.

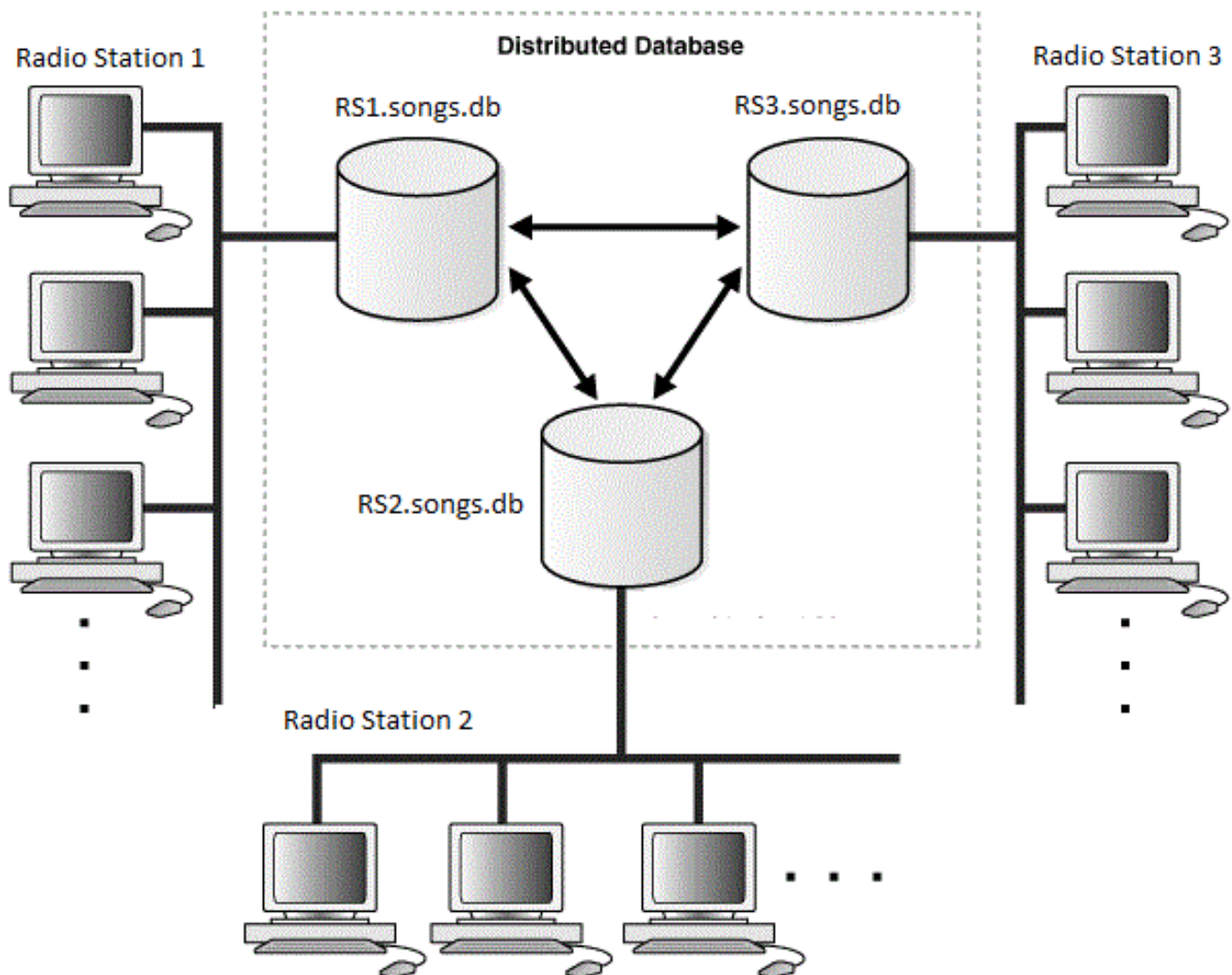


Figure 1.0.0 Distributed Database Solution Architecture [4]

Synchronous replication should be used by Beaconside Radio because it will guarantee that the other systems have a copy of the data, this is ideal due to the radio stations wanting to the same music available to all of them, a disadvantage is the response time is increased. [2]

Non-synchronous replication could be used in the future if Beaconside Radio mergers with radio stations that are not local (exceeded 300km), [3] this seems unlikely because their business aims are to support local organisations / performers. It could also be considered due to Synchronous replication being expensive in comparison.

4.0.0 Cloud Database Solution

4.1.0 Cloud Database Solution Explained

A cloud solution could be a Database as a Service which I would recommend because of not having to install or maintain the database due to it's the responsibility of the provider this is beneficial for Beaconside Radio due to paying for how much they use and not having a database administrator.

The diagram shown below represents an example of a cloud database. The database can be extended easily because you can upgrade your cloud package with the provider when needed.

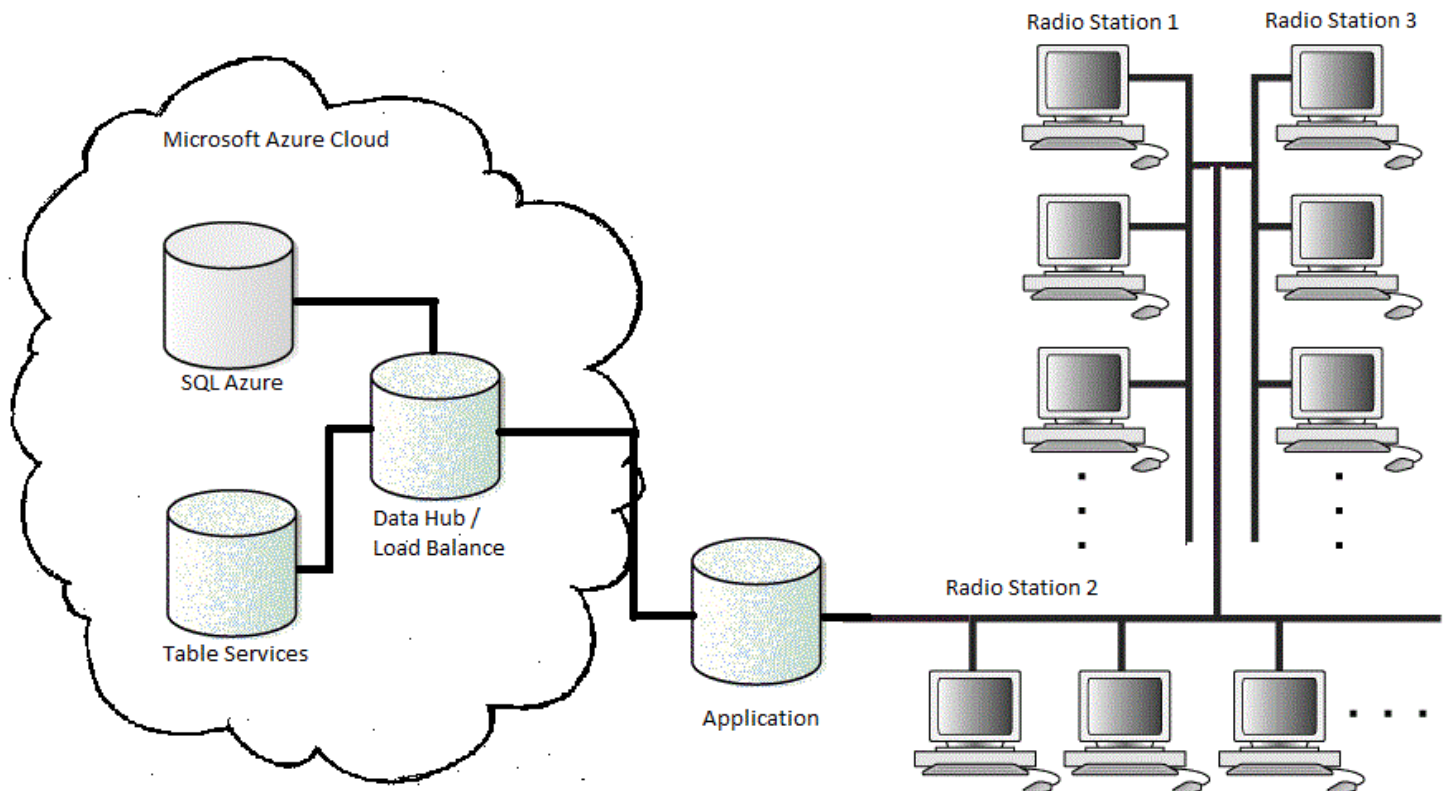


Figure 2.0.0 Cloud Database Solution Architecture

With a cloud database the administrator has a lot more time to focus on the task specific to the business aims and objectives, this will also allow the radio station to take on a volunteer/intern to reduce wage costs, also the cost of a cloud database is a lot less the implementing a distributed database from scratch where most cloud services offer pay by hour, pay by use and subscription based payments. [6]

One of the major concerns about cloud computing is its security where it is up to the hosting company to get the private data and information that might be sensitive or confidential protected, so the choice in cloud provider is paramount so it's best to stick to a well know provider like Google Cloud SQL, Microsoft Azure or Amazon Web Services. A database administrator could increase security by implementing user accounts and object-level security on the server. [6]

5.0.0 Recommendation to Beaconside Radio's Management

5.1.0 Recommendation Explained

The research has shown that many different options are available to Beaconside Radio but in conclusion it all depends on how much Beaconside is willing to spend, I would suggest a cloud database solution using one of the well know providers this will help with any security concerns. A cloud solution also has a reduced cost when compared to a distributed database, where you can use pay per hour in most cases so when the radio station is offline and the database isn't being used you will not have to pay.

Also the cloud provider will be responsible for protecting and maintain the database, this would keep the technical tasks to the provider and away from Beaconside Radio, if any task were required I would suggest that a volunteer or intern was hired who has specialist experience with cloud databases. With a cloud solution the system because very cheap to extend because it doesn't require new software / hardware, you only need to upgrade your package with the cloud provider.

6.0.0 Appendix

6.1.0 Assumption

During the course of researching and recommending a solution for the management of Beaconside Radio the following assumptions where adhered to.

- All radio stations will use the same data model with new system.
- The financial model of the solution is a major factor.
- The Radio station already have basic software/hardware (Desktops, Internet Access).
- Staff will receive appropriate training to use the new solution.
- The solution will support the continuing expansion of Beaconside Radio
- A NoSQL solution will not be suitable due to data volumes not being sufficient.

6.2.0 Word Count

Word Count [998]

7.0.0 Bibliography

7.1.0 Website References

- [1] MongoDB. (2013). *NoSQL Databases Explained*. Available: <http://www.mongodb.com/learn/nosql>. Last accessed 5th Jan 2014.
- [2] Margaret Rouse. (2011). *Synchronous Replication*. Available: <http://searchdisasterrecovery.techtarget.com/definition/synchronous-replication>. Last accessed 5th Jan 2014.
- [3] Margaret Rouse. (2011). *Asynchronous Replication*. Available: <http://searchdatabackup.techtarget.com/definition/asynchronous-replication>. Last accessed 5th Jan 2014.

- [4] Oracle. (2013). *Distributed Database Architecture*. Available: http://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts001.htm#ADMIN12075. Last accessed 5th Jan 2014.

- [5] DINESH THAKUR. (2013). *What are the Advantages and Disadvantages of Distributed Database Management System?*. Available: <http://ecomputernotes.com/database-system/adv-database/advantages-and-disadvantages-of-ddbms>. Last accessed 5th Jan 2014.

- [6] Dawn Altnam. (2013). *Cloud vs. Traditional databases: What's best for your business?*. Available: <http://www.richplum.net/cloud-vs-traditional-databases-whats-best-for-your-business/>. Last accessed 5th Jan 2014.