Improving Data Access Performance Using Coherence Caching in SOA and ADF Application

Kuber Vikram Singh [1] ,Inderjeet Yadav [2]
[1]M.Tech Scholar, Department of Computer Science & Engineering, NGFCET,
Maharshi Dayanand University, Palwal, INDIA.
[2]Assistant Professor, Department of Computer Science & Engineering, NGFCET,
Maharshi Dayanand University, Palwal, INDIA.
E-mail:[1]KuberVikramSingh@gmail.com, [2]yadavcse@gmail.com , www.ngfcet.com

Abstract:- In the Current software applications design, there are multiple calls made to the Data base from various applications like ADF, SOA etc. These data base calls make resources busy between applications and data base. Frequently data base call makes system/Applications slow. So now a days we are dealing with large applications with loads of read and write operations and every call made to the data base will cost more from the performance perspective. As a workaround, We will provide solutions cache, a cache is a component that transparently stores data so that future request for that data can be served faster. Oracle Coherence is a caching strategy so by cache will increase the performance of data fetching and End User will get fast response to his query.

Keywords*: Data Fetching, Data Base, Caching, Oracle Coherence, End User Applications ADF, Middle layer SOA, Oracle Data Base*

*****

1. Introduction

In the current HRDM (Human Resource Data Management) design, there are multiple calls made to the Data Base from various ADF and SOA applications. These calls to the Data Base are being made primarily to fetch the employee, organization and user role information.

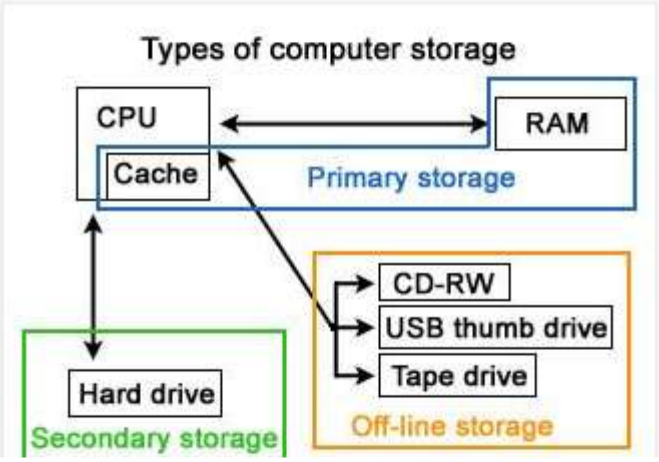


Fig 1.1 Secondary storage

AS–IS Human Resource Data Manager (HRDM)

Current Process ADF End user Applications connect with data base directly for fetching data for end user Query.

Sequence Diagram - ADF

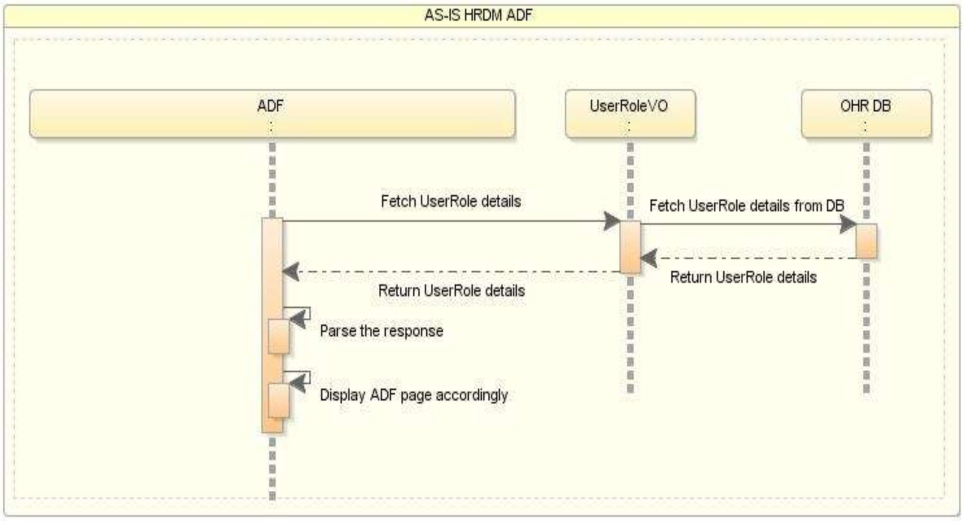


Fig. 1.2 AS-IS ADF Process

Sequence Diagram - SOA

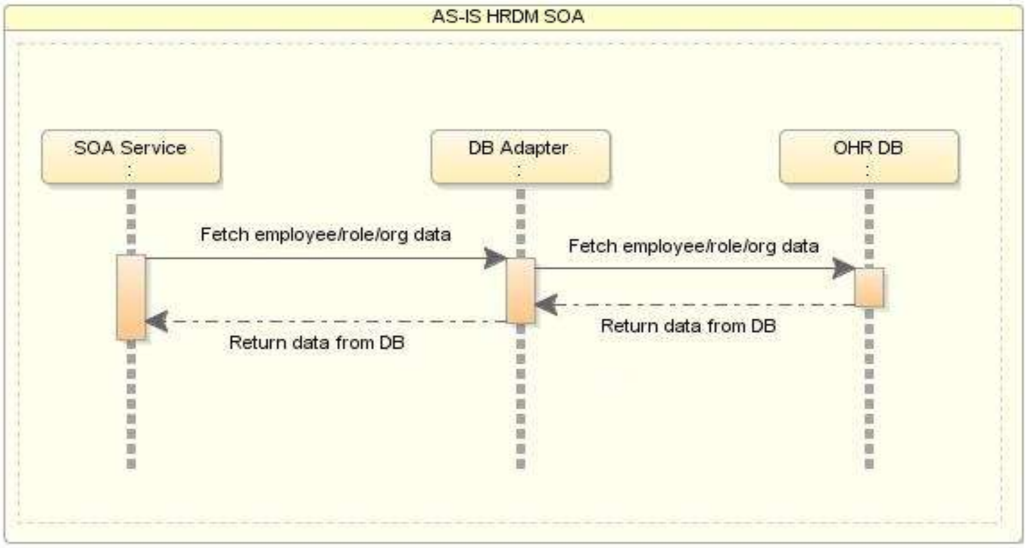


Fig. 1.3 AS-IS SOA Process

2. Background

In the new design, the idea is to cache the frequently fetched DB data, using Oracle Coherence. The integration with the cache would be using java code, which will be further exposed to ADF/SOA applications using an OSB layer. REST based services will be exposed for ADF, and Web Services will be exposed for SOA applications, using Oracle Service Bus.

3. Proposed Work

In the new design, the idea is to cache the frequently fetched DB data, using Oracle Coherence. The integration with the cache would be using java code, which will be further exposed to ADF/SOA applications using an OSB layer. REST based services will be exposed for ADF, and Web Services will be exposed for SOA applications, using Oracle Service Bus.

TO–BE HUMAN RESOURCE DATA MANAGER (HRDM)

In proposed work, ADF end User Applications fetch data from Primary memory so it saves time and end User gets result in very less time compared to the earlier method because it is not fetching data from the secondary memory.

Sequence Diagram - ADF

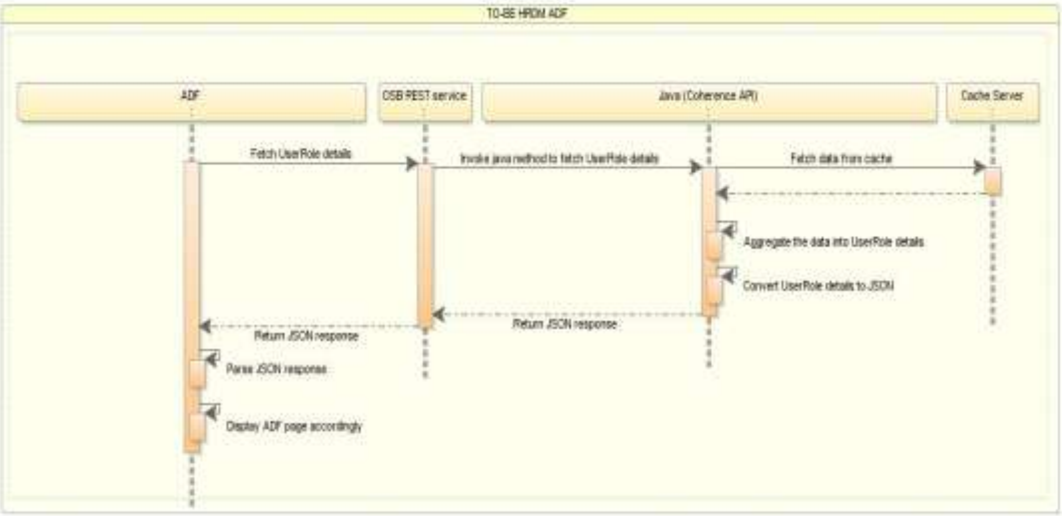


Fig. 3.1 TO-BE ADF Process

Sequence Diagram - SOA

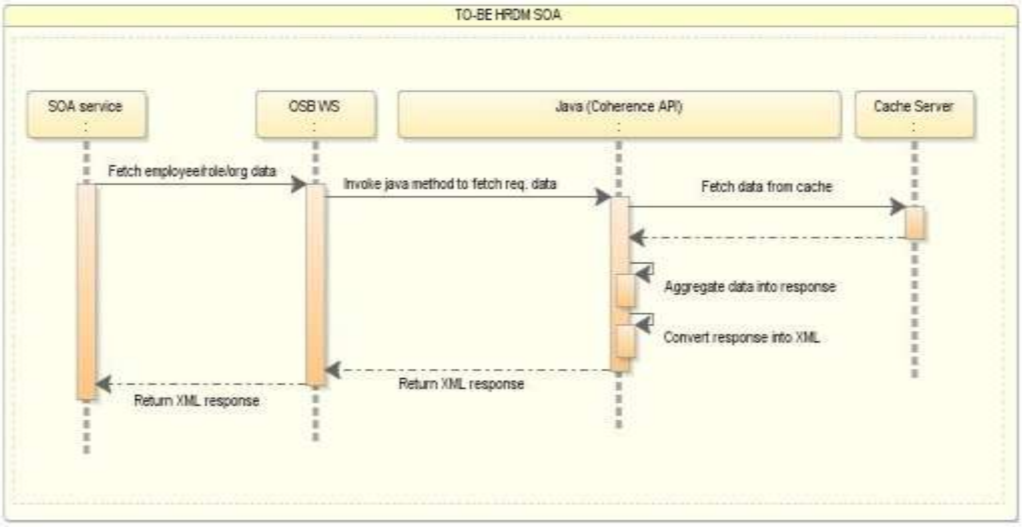


Fig. 3.2 TO-BE SOA Process

BATCH PROCESS FOR DATA TRANSFER FROM DATA BASE TO CACHE:

For transferring data secondary memory (Data Base) to Primary memory Cache, we have batch process schedule in the system for this.

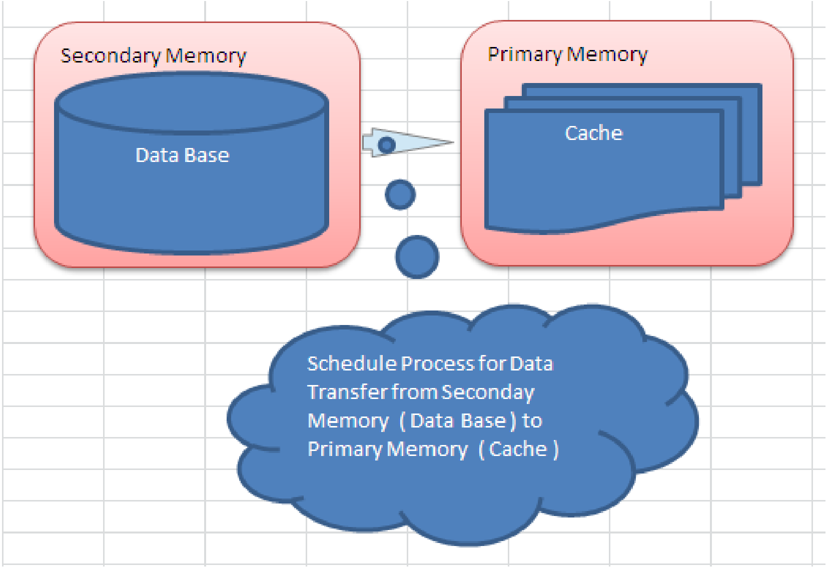
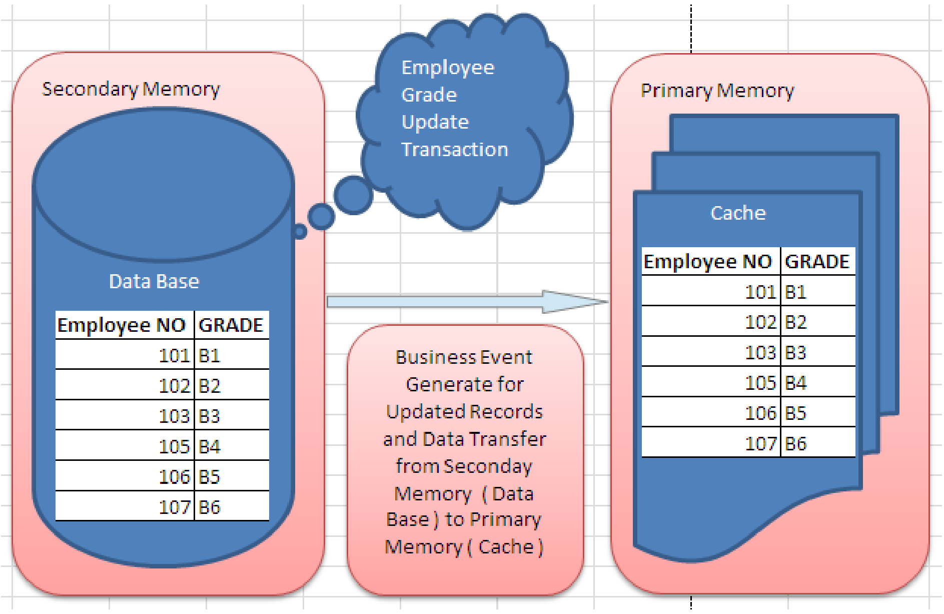

Secondary Memory
Data Base
Primary Memory
Cache
Schedule Process for Data Transfer from Seconday Memory (Data Base) to Primary Memory (Cache)

Fig. 3.3 Batch Process for Data Transfer

REAL TIME DATA UPDATE PROCESS FLOW:

Primary memory (Cache) should have same data as per secondary memory (Data Base) so if any records update at data base level then Business Event generate and same records it transfer from secondary memory (Data Base) to Primary memory (Cache)


Secondary Memory
Data Base
Employee Grade Update Transaction
Business Event Generate for Updated Records and Data Transfer from Seconday Memory (Data Base) to Primary Memory (Cache)
Primary Memory
Cache

Employee NO	GRADE
101	B1
102	B2
103	B3
105	B4
106	B5
107	B6

Employee NO	GRADE
101	B1
102	B2
103	B3
105	B4
106	B5
107	B6

Fig. 3.4 Real Time Data Update Process Flow

Better scalability and performance – combination of replicated and partitioned topologies

- •Pros:
  - •The same as partitioned
  - •Faster access
- •Cons
  - •Updates costs more
- •Primary Use:

•Big imdb with continuous data access.

4. Conclusion Future work

Using Data Cache, It increases query response time. Better Processing and lower convergence time is always a challenge in Data base and Query Optimizations of Cache Data Base is always required.

References

[1] https://docs.oracle.com/cd/E13924_01/coh.340/e13819.pdf
[2] https://docs.oracle.com/middleware/1212/coherence/COHTU.pdf
[3] https://docs.oracle.com/cd/B14099_19/caching.1012/b14046.pdf
[4] https://docs.oracle.com/cd/E23943_01/dev.1111/e15866.pdf
[5] https://docs.oracle.com/middleware/1212/coherence/COHAG.pdf
[6] http://docs.oracle.com/cd/E24290_01/coh.371/e22839.pdf