Trinamix

Splitting large volume BIP Report output into multiple output files

Whitepaper By - Prasad B Chavan

Trinamix

Table of Contents

Introduction

Overview

We were working with the energy and water resource management client on Supply planning implementation project. As part of that there were some reports which need to be migrated to cloud "as-is", since reports are built on plan specific data and then fed to downstream applications.

Initially the solution looks simple, as we made required changes in report sql at cloud, then through SOAP API report is submitted and generated csv file put at SFTP, and using middleware file loaded to the EBS side staging table.

But later we started observing the spike in supply plan data volume. Result of this, we started facing issue while inserting data in to custom staging table. So, we decided to create multiple reports based on date horizon. Since the plan data is incremental, we started facing difficulty to determine report count that is needed for a single plan. Also creating multiple report isn't robust solution.

Alternative

BICC was the option we had, to pull high volume data and recommended by oracle as well, but its standard view object does not contain the important columns that required in report.

Solution

As part of solution, we decided to use BI bursting by sub setting the data to smaller data set using ORA_HASH standard oracle function, as shown in below solution diagram.

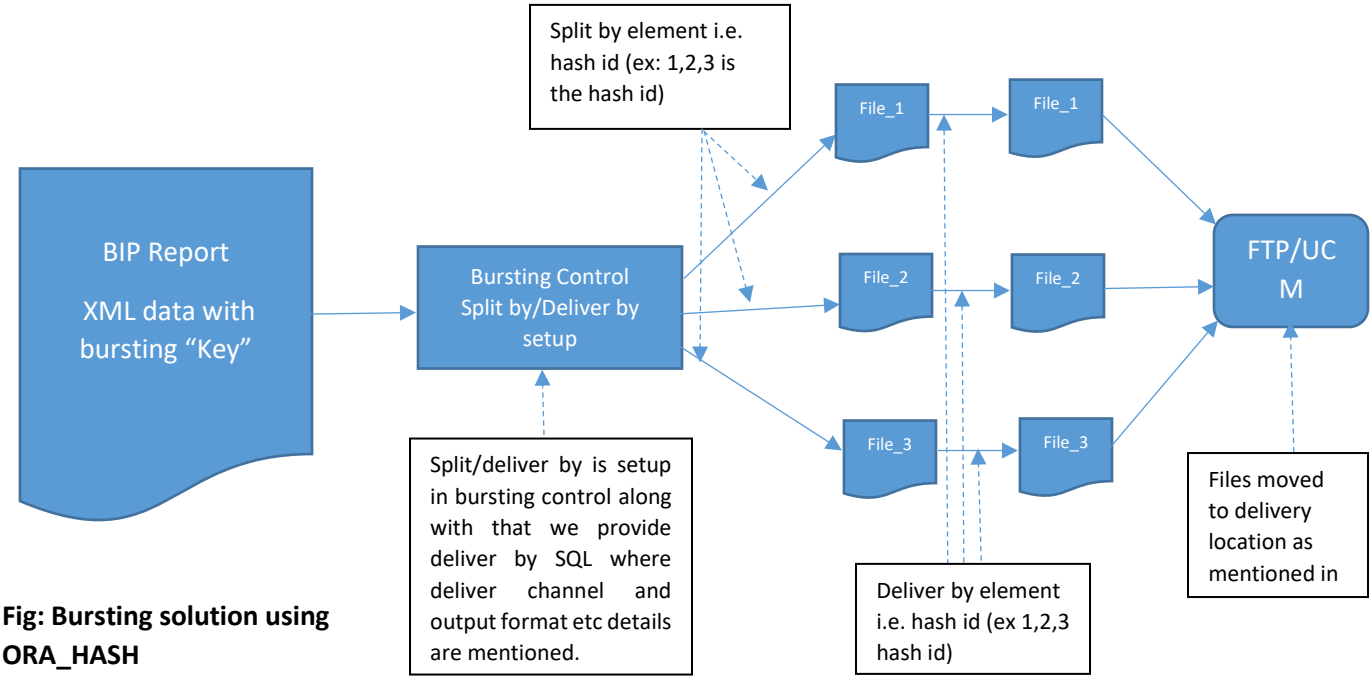


Fig: Bursting solution using ORA_HASH

ORA_HASH is the function, generates the hash id (integer value) for given expression, which we used to divide the data into subsets in our reports. BI Bursting is using the hash id to split and deliver the data at the FTP.

Bursting is a process of splitting data into blocks, generating documents for each block, and delivering the documents to one or more destinations. The data for the report is generated by executing a query once and then splitting the data based on a "Key" value. We're using HASH_ID as key and it is used as common split by and deliver by element. Following steps need to perform to implement the solution:

- Create report and add ORA_HASH in bursting sql by providing expression (expression here we are passing as 'column name' of table which using in SQL) and buckets i.e. number of data set that we want to generate.

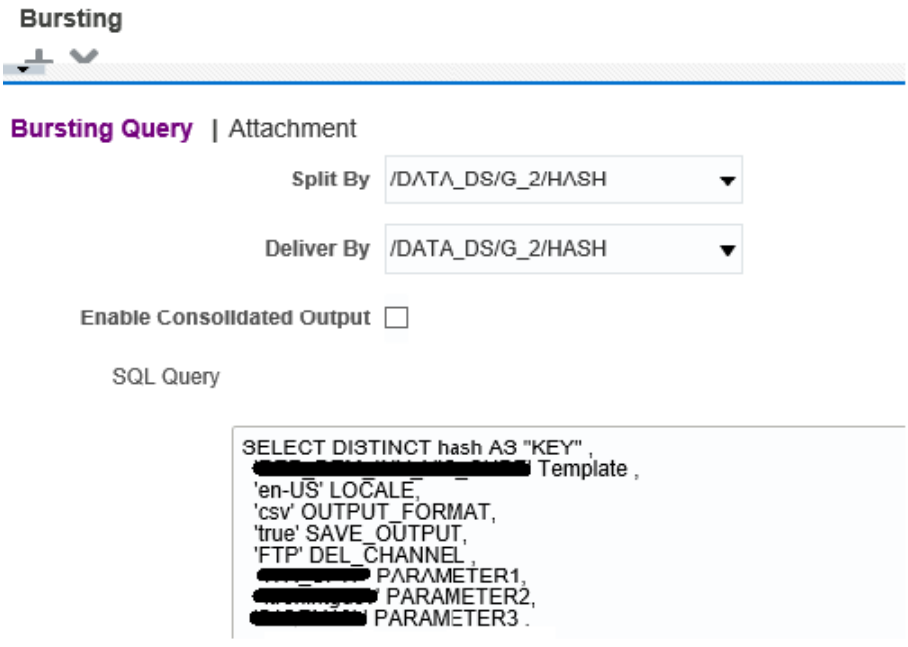
Viz. ORA_HASH (demand_id,10)

Expression **no. of buckets**

demand_id is the column name used in report SQL used as expression and bucket is 10 means data will divided into 11 buckets range from 0 to 10 which we called as HASH_ID.

```
select plan_id,
inventory_item_id,
ora_hash(dem.demand_id,10)  hash
FROM msc_demands dem,
  msc_system_items msi,
  --msc_key_customer kc
  msc_full_pegging mfp
WHERE msi.plan_id        =NVL(dem.base_plan_id
AND msi.new_plan_id      = -1
```

- Create bursting definition within the report, mention split by and delivered by element, in deliver by SQL add required bursting parameter like deliver channel, Key, output file name etc.



- Run the report and check the output at delivery location i.e. FTP/UCM.

Pros and Cons

- **Pros**
  - Can be used to interface any high-volume data using BIP report to downstream application by sub-setting data set using ORA_HASH.
  - Improved BIP Report performance in terms of runtime by 20-30%.
- **Cons**
  - Downstream application has to be prepared to handle the multiple files.

Conclusion

Using BI bursting and ORA_HASH we can split any large csv/excel BIP report output into multiple files.

About Author:-

Prasad Chavan is working with Trinamix Inc. as a Consultant having good experience Over 7+ years in implementing, developing and supporting VCP Modules for discrete manufacturing and worked with client in different geographies. Worked as Onsite lead for EBS SCM and VCP applications for UK based client. Focus area ASCP, Fusion Supply planning and analytics, OTBI. Completed BE in Computer Engineering and PGD in Operations Management.

www.trinamix.com