Oracle BAM 11g Best Practices v4

Supported version: 11.1.1.2 and above

Introduction

This document provides information about the Oracle BAM best practices for design and deployment. Oracle BAM is a real-time monitoring tool that allows business managers and executives to manage their operations in real time. This application allows user to not only monitor, but also take corrective action on the information presented in the real-time business dashboards.

The following sections detail the best practice recommendations for any Oracle BAM application. Please note that this is a working document and as more customer implementation designs as well as requirements are available, we will continue adding to this document. Following this document does not preclude a design review as well as performance testing based on the dashboards/ volume/ schema design implemented .

Dashboard Design and Deployment

1. Ensure that a Dashboard is designed for a specific user group. Do not try to create an "all-purpose" dashboard.
2. Allow the user to understand the health of the operations at a glance, and then provide drill hierarchies to perform root cause analysis.
3. Limit a dashboard to 4-6 views per report. Because Oracle BAM is a push-based technology that streams updates to the dashboard, try to limit the number of views in the open report for optimum performance.
4. Use Report prompts and parameters to filter data before report loading. This is especially useful when there is a large amount of information available that may not be relevant for the business user, e.g. a full month of data when all they want to see is last 2-5 days.
5. Creating URLs with "Copy shortcuts" in Oracle BAM Active Studio can be used to save time in opening a dashboard.
6. Make effective use of Action buttons and action forms to ensure the end user can take corrective action on the dashboard.
7. Try to use summary /aggregate views such as bar charts, pie charts, collapsed lists. This way only the aggregates need to be maintained in the cache.

8. List views can have a performance impact. Do not show too many lists on the report.
9. Use lists with caution! Oracle BAM is used to providing a real-time status of the business. Listing hundreds/thousands of records in a list view will impact performance. If there is a lot of data for a list view, allow the user to provide filter criteria (using a prompt) for the list rather than display all records by default.
10. Use the Oracle BAM Dashboard view to put together common reports that may be used by multiple user groups. For example, sales folks need two views while managers need the two common views plus a regional view. An Oracle BAM dashboard allows the user to bring in the common the two view report in both dashboards, while the manager dashboard will have an additional view.
11. Cross Tab Summary and similar views. Keep in mind that the performance for the views is impacted based on the number of groupings and records, because real-time updates need to be applied to the view. If a Cross Tab has 5 fields to group by, it will be significantly slower than two fields. Once again, it is important to keep #1 and #2 in mind during design.
12. Oracle BAM Web applications cache lots of files on the browser, so it is recommended that the user does not clear the cache unless necessary.
13. The user can make use of the active data interval setting to suit their needs, as high frequency updates come with a performance cost.
14. Oracle BAM Alerts are the convenient way to receive report snapshots as emails.
15. Use pre-packaged demos for design guidelines.
16. Use recommended browser versions for better performance. (IE 7 and 8 for 11.1.1.2.0).
17. Surface Prompts are extremely expensive in performance . Also, ensure that if there is a default value set to the prompt, it will be relevant even with data archiving /purging routines. Else, this will impact report performance
18. Use BAM for operational monitoring where you want to take action in short term e.g. average time for processing over x amount of hrs or days/ faulted count of instances which could turn to xxx amount of orders that cannot be processed and need attention, orders in jeopardy since it hasn't reached step 4 and only 2 days left from promised date.

Schema

19. Oracle BAM is used for real-time monitoring needs by business managers. The Oracle BAM ADC should not replace the customer's data warehouse needs. Oracle BAM should keep only recent data in the schema. If required, the customer can archive the granular data and only keep aggregate data in Oracle BAM for a specific period of time.

20. Oracle BAM supports star schema design. It supports one level of lookups, i.e. from one table to another. However, it does not support snowflake designs.
21. It is recommended that Dimension and Fact tables (data objects) be separated. However, if your dimension table is growing at a fast pace, or at the same pace as the fact table, it is a good idea to not separate these two. For example, if Order table and Invoice table are growing at the same rate and have lookups from one to the other table it is better to keep this as a single table.
22. It is better to handle calculated fields at data object level, rather than at the report level. For example, sales amount *1.2 could be an extra field in the data object rather than creating a calculated field in the report.
Note: The Calculated field expression builder has syntax verification in the Active Studio and not in Architect. Use copy and paste into Active Studio calculation expression builder to verify syntax if required.

23. Create indexes for key fields. For example if you are using BPEL sensors to do insert/upsert/update functions within a process, create the index on Oracle BAM schema on the keys. Create Keys with Integer format.
24. Indexes can also be created for helping in report viewing, i.e. the fields that the user is going to search on in the reports. However, do not create too many indexes.
25. External Data Object: If the requirement is to filter on a calculated field from an external data object, create a VIEW on to the table with the calculated field. Use the resulting VIEW for creating an external Data Object.
26. BAM is a reporting tool -as with any other reporting tool, BAM should have its own DB instance

Data Archiving and Purging

27. Keep **only** the necessary data in your Data Object. Any data that is relevant for real-time operational monitoring and taking actions is a candidate for storing in the data object for the reporting.
28. Do not keep older data for historical purposes in the same data object.
29. If you need the historical data available in Oracle BAM, try to either store aggregates in another Data object, or transfer the data to another data object that is only called upon for specific reporting.
30. Oracle BAM has an alert configuration for purging data from the Data object. This can be scheduled for data older than a defined time period. It is recommended that the purge be scheduled at non-peak hours, e.g. nightly.
31. If it is a requirement to archive the data prior to purging, then use ODI with Oracle BAM 11g, or Enterprise Link Plans in Oracle BAM 10g, to extract the data from the Data Object and store it in the data warehouse/table. Oracle BAM

allows external data sources for reporting purposes without bringing in the data into Oracle BAM.

32. ODI should have its own server . However, it should be co-located with the BAM server.

Scalability and Performance

Oracle BAM High Availability solutions are available in SOA Document library. http://download.oracle.com/docs/cd/E17904_01/cross.htm

33. Oracle BAM Server (ADC) is a singleton and available in Active /Passive Configuration. Oracle BAM Web can be deployed in Active-Active configuration. It is recommended that a sizing exercise is accomplished as part of the Oracle BAM design and system testing for a specific scenario.
34. Oracle BAM deployment sizing guidelines are provided as a technical note on the Oracle BAM section of the Oracle Technology Network Web site.
35. It is recommended that dedicated machines be used for Oracle BAM (separate Oracle BAM Server and database machines, and others based on install size) because Oracle BAM is a reporting tool. If the solution includes BPEL, use separate machines for BPEL and its storage to avoid contention.
36. For high availability solutions, make sure that your servers are on the same subnet.
37. Ensure that the application context is the same for Oracle BAM Web services if the Web services interface is used for pushing data in Oracle BAM.
    The Oracle BAM Web services use different a application context than **OracleBAM**, so adding the below lines along with WSM-PM definition solves the issue.
    ```
    <Location /OracleBAMWS >
         SetHandler weblogic-handler
         WebLogicCluster BAMHOST1VHN1:9001,BAMHOST2:9001
      </Location>
    ```
38. The Oracle BAM Enterprise Deployment guide should be referred to for deployment guidelines:
    http://download.oracle.com/docs/cd/E17904_01/soa.htm
    http://download.oracle.com/docs/cd/E17904_01/core.1111/e12036/toc.htm

    Performance and Tuning Guidelines are listed here:
    http://download.oracle.com/docs/cd/E17904_01/cross.htm

    http://download.oracle.com/docs/cd/E17904_01/soa.htm

Disaster Recovery Guide is listed here:
http://download.oracle.com/docs/cd/E17904_01/doc.1111/e15250/recommendations.htm#CHDDHHJG

Real Time Data sources for Oracle BAM

39. There are 4 ways to push data in Oracle BAM. Please note that we recommend that the DO should be updated via one common channel as this is the most efficient and optimized implementation for updating the DO. It is OK to update the DO from multiple projects as long as they are not writing to the same DO record concurrently. **We strongly recommend against updating the same DO record via multiple sources (e.g. multiple EMS, or multiple BPEL projects etc.)**

(1) Oracle BAM Adapter – JCA adapter available in 11g. This is also a first class citizen of SOA composite. This is the recommended approach for sending data to Oracle BAM for BPEL and other applications certified with this adapter.
   i. The adapter supports RMI and SOAP (HTTP and HTTPS) protocols.
   ii. BPEL 11g uses Oracle BAM Adapter to send data into Oracle BAM via the Oracle BAM sensor actions.
   iii. This is **not guaranteed messaging**. All batches are stored in virtual memory. If there is a requirement for guaranteed messaging, please use JMS sensors in BPEL to send information to Oracle BAM via Enterprise Message Sources.
   iv. Enable batching to keep the BPEL-Oracle BAM sensor action communication asynchronous. This also applies to the Oracle BAM Adapter use at the composite level. Oracle BAM 11g allows configuration of batch size, time-out, retry, etc.
   v. Select the Monitor perspective in Oracle JDevelepor BPEL Designer (top right drop-down with BPEL as default) to create sensors /monitors to allow process monitoring.

(2) Web services – Each data object is exposed as a Web service in Oracle BAM.
   i. BPEL 10.1.3.x uses the Web services route for sending data into Oracle BAM using Oracle BAM sensor actions.

      ii.   This is not guaranteed messaging. All messages are stored in virtual memory. Enable batching in BPEL – Oracle BAM sensor communication to maintain asynchronous communication between the two applications.

      iii.   The Oracle BAM ICommand utility is available as a Web service.

(3) JMS – Oracle BAM 11g has native JMS connectivity. It has the ability to listen to any JMS (1.1 or above) queue or topic. This also allows durable subscription. This is a recommended mode of communication for high volume messaging scenarios.

- JMS supports XSLT transformations. Oracle BAM requires a flat schema where the XSLT can be used for nested messages. Please note the XSLT requires more resources for processing in high volume scenarios.
- Enable EMS message batching: This groups the messages in one single payload and saves on network round trips.
- As the JMS message is pulled from the JMS topic, if the record cannot be processed, it will not be written back to the same topic automatically. There are error configurations in EMS (post 11.1.1.3.0) that should be taken advantage of to capture records that are rejected by the EMS.
- Only 1 EMS should be used to write to 1 Data Object. This will sequence of updates as well as avoid any contention during the update of the same record via multiple sources.
- Recommend EMS to be configured with transactions disabled (11g)
- Do not recommend transformations on EMS especially in high volume scenarios. It is much better to transform the message in BPEL (if that is being used) or the source system rather than overloading the reporting tool for transformation and then updating in real time.
- If you send null values for a column value into BAM, it will overwrite the existing value. If you want to preserve the value for a column ,then do not send a null – infact suppress the null value tags in the message.
-

(4) Oracle Data Integrator (ODI) – Use ODI (10.1.3.5 recommended) to extract data from any compatible source and push it in Oracle BAM using the Oracle BAM Knowledge Modules. This tool also is used for archiving Oracle BAM data. Change Data Capture can be supported using ODI integration with Oracle BAM.

(5) Use Oracle BAM ICommand command line tool for exporting and importing Oracle BAM artifacts.

(6) Ensure that the Data Object field size is suitable to the payload size. Recommend field sizes less than 1000 characters, if possible

(7) If there are multiple BPEL processes /composite writing to Oracle BAM, it is a good practice to have all the various processes raise EDN events (or write to a JMS topic) and then have one single composite with BPEL process write to BAM.

Oracle BAM Data Control

40. Oracle BAM data control instance sharing is not supported. When two or more ADF Faces components must display the same data, and are bound to the same Oracle BAM data control definition, make sure to wrap each ADF Faces component in an ADF task flow, and set the Data Control Scope to *isolated*. See the *Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework* <http://download.oracle.com/docs/cd/E15523_01/web.1111/b31973/toc.htm> for more information. When using an Oracle BAM data control with `setParamters` inside task flow, the user needs to pass the parameter to the task flow and ensure that the method gets called before the query call.

41. Deployment of ADF applications with Oracle BAM data control(s) on a non-integrated (not installed and managed by Oracle JDeveloper) development mode Oracle Weblogic Server can be simplified by passing `jps.app.credential.overwrite.allowed=true` on Oracle Weblogic Server startup.
On Microsoft Windows:

    set JAVA_OPTIONS=%JAVA_OPTIONS% -Djps.app.credential.overwrite.allowed=true

    restart WLS and redeploy the application

Other useful links

- http://www.oracle.com/technology/products/integration/bam/index.html; Refer to the tech notes and tutorials section.

Questions & Clarifications:

If you have any comments or need additional information, please communicate through the Oracle BAM forum at: http://forums.oracle.com/forums/forum.jspa?forumID=252