Agenda

Company Profile
Challenges
Requirements
Approach
Infrastructure
EM Implementation Details
Benefits
Common Commands
Q/A
AT&T

• AT&T is a leading provider of wireless, Wi-Fi, high speed Internet, and voice services
• 90.1 million wireless subscribers
• More than 129,000 Wi-Fi hotspots around the globe
• The nation’s fastest mobile broadband network
• AT&T’s global network handles nearly 19 petabytes of traffic on an average business day
• 2.5 million AT&T U-verse TV subscribers
• 100 percent of Fortune 1000 companies are AT&T customers
• In 2010, again ranked among Fortune’s 50 Most Admired Companies
• Global headquarters located in Dallas, Texas
One of the DBA support teams in AT&T managing databases.

- 2000+ ORACLE DBs
  - Multiple Versions
  - Features
    - RAC
    - Data Guard
    - Golden Gate
    - Streams
    - Flashback
- 60+ DBAs
  - Multiple sub teams
Challenges

Database Management and Diagnostics: Wide range of ad hoc tools in use. Management complexity.

Database Monitoring: Multiple home grown custom monitoring solutions developed over the years.

Database Scripts: Complexity with script rollout, updates and version changes.

Database Version Complexity: Hard to keep up with changing data dictionary views in newer Oracle versions.

Database Diagnostics: Growing performance and availability requirements for our databases and existing tools cannot keep up with them.

New DB Features Support Complexity: Supporting new DB features involved creating scripts, custom monitoring solutions and building tools.
Requirements We Set for EM

Provide Ease of Database Management
- Perform all database management duties using the tool
- Manage new database features with ease

Database Troubleshooting and Performance Tuning
- One common tool for enterprise to troubleshoot database performance issues

Monitor All Databases
- Provide a monitoring solution that is easy to manage and will scale well to meet all of our requirements

Database Build Automation
- Ability to provision Oracle database software and automate the database build process
Approach: Road to EM 10.2.0.4

POC
- Grid stability
- Agent scalability
- EM monitoring capabilities

Design & Development
- Design EM solution
- Develop custom solutions for EM agent deployment, availability and additional metrics for monitoring

Production Implementation
- EM with DR implementation
- Agent and monitoring deployment
EM POC results showed that EM 10g can meet our requirements, but custom work was still needed on the following areas:

- Agent mass deployment
- Agent availability (automatic start/stop)
- User defined metrics to plug monitoring gaps
- Automate target configuration to the appropriate DBA teams
Deploy the latest EM version available at that time which was 10.2.0.4. Work with Oracle to identify all the patches required for a stable environment.

Automate agent deployment using cloning technique.

Since agent availability is critical for monitoring, develop scripts to auto start/stop agent during server reboot, database failovers and to restart agent when they are down for other reasons.

Add additional monitoring using user defined metrics to our requirements. Define and deploy monitoring metrics through templates.

Use EM groups for managing target ownership. Develop custom process using EMCLI to manage groups based on our internal database inventory data.
EM Implementation – Time Capsule

- Three major phases
- Proof of concept in 2007
- Production deployment in 2008
- Monitoring implementation in 2009
## Key EM Features Used

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle Software Cloning</strong></td>
<td>• Deploy a standard and fully patched EM agent software across all grid targets.</td>
</tr>
<tr>
<td><strong>EM Groups</strong></td>
<td>• Target ownership, pushing out monitoring templates, notification, dashboards, ease of management</td>
</tr>
</tbody>
</table>
| **Monitoring Templates** | • Target monitoring metrics and policies management  
• Used in conjunction with groups                                                                                                           |
| **Notification**         | • OS & SNMP notification methods to page/email alerts out to the appropriate recipients within the DBA team  
• Email repeat notification feature                                                                                                         |
| **UDMs and UDPs**        | • User Defined Metrics and User Defined Policies are used to meet monitoring needs related to database administration, performance, backups, Golden Gate, compliance programs like SOX, PCI |
| **EMCLI Commands**       | • Extensive use of EMCLI commands for target configuration, EM group and template management, agent management, password changes |
Agent Installation and Configuration - Implementation

Agent Install
- Copy the Agent Clone Software
- Run runInstaller command to clone the Agent Home

Target Configuration
- Agentca -f command to discover target
- Emcli modify_target command to set password

Setup Monitoring
- Push appropriate metrics using monitoring templates based on the target type
- Emcli apply_template command to push templates

Configure EM Groups
- Add the newly added targets to appropriate EM groups and EM roles
- Emcli modify_group and emcli grant_privs command to configure groups and roles
EM Agents – Key for a Successful Implementation

EM Agents are set up to meet the following requirements:

**Performance**
- Monitor agent operations (trace files, log files) from time to time. Review metric collection errors, metrics extending beyond interval errors. Cleanup of agent log files on periodic basis.

**Availability**
- Auto start/stop script integrated with VCS cluster software where applicable.
- Tracking agent non-availability through EM repository views and starting agents on demand.

**Stability**
- Monitor agent operations (trace files, log files) from time to time. Review metric collection errors, metrics extending beyond interval errors. Cleanup of agent log files on periodic basis.

- Standardization of agent software. Only 10.2.0.4 and above versions with Oracle recommended patches are deployed in our environment.
Monitoring Solution Through EM – Key Components

- **Metrics & Policies**
  - Standard Metrics
  - User Defined Metrics
  - Metric Thresholds
  - Policies
  - User Defined Policies

- **Monitor**
  - Monitoring Templates
  - Agents
  - Targets (Database, Host, Listener)

- **Notify**
  - OMS
  - Notification Rule
  - Notification Methods

We used standard metrics, UDM, custom metric thresholds, UDP, monitoring templates, notification rule, OS and SNMP notification method for our monitoring solution
# Database Monitoring – Challenges with Out of the Box Metrics

## Issues:

<table>
<thead>
<tr>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrics for conditions that were not appropriate for some of our databases</td>
</tr>
<tr>
<td>Metrics that produced too many alerts</td>
</tr>
<tr>
<td>Metrics that didn’t exist for conditions that are deemed as required for our environment</td>
</tr>
<tr>
<td>Bugs with some metrics that are based on the database server generated alerts in 10g</td>
</tr>
</tbody>
</table>

## Solution:

<table>
<thead>
<tr>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable the metrics where possible. If there is a dependency on other metrics, then nullify the thresholds</td>
</tr>
<tr>
<td>Adjust thresholds, number of occurrences to reduce the quantity of alerts</td>
</tr>
<tr>
<td>Develop User Defined Metrics for missing monitoring conditions</td>
</tr>
<tr>
<td>Work with Oracle to resolve bugs. If not possible work around the issue with UDMs</td>
</tr>
</tbody>
</table>
Database Monitoring – Standard and UDM Metrics Usage

Out of the Box Metrics

- Used for database, listener and host targets
- Only required metrics are used after thorough testing for reliability
- Thresholds, number of occurrences used. Frequency never adjusted as per Oracle recommendation
- Metrics used include: Availability, performance, alert log with special filters, space, RAC, data guard

User Defined Metrics

- UDMs are only used for database targets
- UDMs used for conditions that cannot be met with standard metrics. Most UDMs are against data dictionary views
- UDM thresholds, frequency are carefully determined to make sure we don’t impact agent and database target performance
- Metrics usage includes: Performance, custom lock monitoring, RMAN backups, scheduler jobs, table space and Golden Gate monitoring

© 2010 AT&T Intellectual Property. All rights reserved. AT&T and the AT&T logo are trademarks of AT&T Intellectual Property.
Database Monitoring - User Defined Metrics – Key Usage Requirements

User Defined Metrics is a powerful EM feature that facilitates adding additional metrics to meet monitoring requirements.

Key Points about UDM

- UDM can only return two columns (key and value)
- In a two-column UDM the first column is the key
- Change of key triggers a clear notification of the previous key record and a new notification for the new key record
- UDMs can only be of a particular type (number or string) and the type is based on the value column
- UDMs requires login credentials to the database
- Can be pushed through templates
Compliance: Standard and User Defined Policies Implementation

Compliance for security is becoming more and more important for database administrators. EM provides standard policies for security and option to create custom one using User Defined Policies.

Enable Policies Related to Security
- We reviewed the available metrics and chose the ones that met our requirements
- Use monitoring templates to enable and disable policies

Create User Defined Policies
- Built User Defined Policies to meet our internal security and SOX,PCI controls related to databases

Reports for Policy Violations
- Created custom reports for policy violations based on repository views to meet the requirements

About User Defined Policies
- 10.2.0.4 allows UDPs to be created using EM packages. 10.2.0.5 provides a GUI screen in EM for UDP creation
- Follows a two step process. Create UDM first and then associate the UDM metadata to create the policy
Monitoring – Lessons Learned from Our Implementation

Plan Your Metrics
- Evaluate and use metrics that are applicable and meets your requirements
- Test your metrics and create baseline metric thresholds based on DB profile (batch, OLTP, mixed) and build monitoring templates based on them

Monitoring Templates
- Use monitoring templates to push out metrics to targets
- If few metrics requires change, consider creating a temporary template
- Build monitoring templates for each target type

User Defined Metrics
- Use User Defined Metrics for cases where standard metrics are not available
- Plan and develop your UDM carefully knowing all the restrictions with using them
## Monitoring – Lessons Learned from Our Implementation

<table>
<thead>
<tr>
<th>Metric Collection Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Remember to track and get notified for metric collection errors. They happen for various reasons (password issues, collection running a long time, bug) and failing to rectify them could result in monitoring failures</td>
</tr>
<tr>
<td>• Query the repository if required to identify these metric collection errors, if tracking them through EM screens is not an option</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Monitoring and Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Monitoring and notification are independent in EM. You can monitor all, but notify only a few. Leverage this feature effectively</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repository Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>• There are repository views that can help provide all the metric information. This feature comes in handy when there is a requirement to compare and validate metrics across large number of targets</td>
</tr>
</tbody>
</table>
EM Notification Implementation

Notification/alerting requires carefully planning in the overall database monitoring strategy. Notification challenges we faced and how we solved them:

**Problem**

- 10.2.0.4 had very little customization available for email notification method
- Our teams required more information about our databases from our inventory records and this required customization
- Requirement to send alerts to different address based on warning and critical thresholds
- Cannot utilize the default schedule

**Solution**

- Used EM OS Notification method as the primary alerting mechanism.

**Customization**

1. EM OS Notification method calls a script in our OMS servers which in turn performs the alerting functionality. EM passes alert information as OS variables and we deliver the alert with formatting, additional information and to the appropriate recipients based on the target name
2. In addition to OS Notification we also use SNMP traps, custom alerting from repository views to meet additional alerting requirements
3. EM 10.2.0.5 and above provides a notification customization feature for the email method
4. EM 10.2.0.5 and above provides repeat notification capability for all methods
Lessons Learned from Our Implementation

Plan Your Metrics

- Evaluate and use metrics that are applicable and meet your requirements
- Test your metrics and create baseline metric thresholds based on DB profile (batch, OLTP, mixed) and build monitoring templates based on them

Monitoring Templates

- Use monitoring templates to push out metrics to targets
- If few metrics requires change, consider creating a temporary template
- Build monitoring templates for each target type

User Defined Metrics

- Use User Defined Metrics for cases where standard metrics are not available
- Plan and develop your UDM carefully knowing all the restrictions with using them
Other Useful Customization: Alert Log Filtering

We came up with a custom alert log filter expression that will only alert for ORA errors that requires DBA’s immediate action.
Post Go Live: Issues
Some of the key issues we addressed after go live:

OMS Performance Tuning:
- Apache HTTP parameter tuning to handle more connections
- Loader backlog: Increase OC4J processes to handle concurrent loader files to avoid backlog

Repository Tuning:
- Increase job_queue_processes parameter to support parallel EM task processing
- Increase redo log size to avoid frequent log switches
- Running repvfy utility on regular basis and take actions to clean out stuck notifications
Post Go Live: Issues

Some of the key issues we addressed after go live:

Agent Issues:

• Agent crashing due to patch conflict with 10.2.0.4 database version patch - resolved by applying the right patch combination

• Agent leaving orphan database connections – resolved by a combination of patching and a housekeeping task to bounce agent prior to hitting that condition

• Missing host performance information in HPUX platform – resolved by patching

• Metric collection errors on standby databases – fixed in 10.2.0.5 agent
Post Go Live: Issues

Some of the key issues we addressed after go live:

Database Monitoring Issues:
• EM Dictionary queries running longer – resolved by collecting periodic dictionary statistics
• Tablespace monitoring inconsistencies – workaround by creating UDMs

Other Issues:
• Load balancer connectivity issue – resolved by LB setting
Supporting EM Infrastructure – Ongoing Support

Some of the key tasks we perform on a regular basis to support this infrastructure includes:

- **Daily health check**: Make sure all targets are running without any issues. Investigate collection errors, pending status state and review performance alerts related to OMS, OMR and agents.

- **Target discovery**: We run into target discovery issues from time to time which requires manual intervention.

- **Agent upload problems** due to connectivity issues.

- **Running repvfy** and taking care of any issues reported.

- **User privilege management** by super administrator.
Q & A