

The Importance of Application Level Monitoring



<http://www.IT-Checklists.com>

Oracle® and Oracle-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation, Inc. in the United States and other countries.

ITIL ® is a Registered Trade Mark, and a Registered Community Trade Mark of the Office of Government Commerce, and is Registered in the U.S. Patent and Trademark Office

CobiT ® is a Registered Trade Mark of ITGI (IT Governance Institute) and ISACA (Information Security Audit and Control Association)

Mercury Consulting is independent of those companies.

Conventions

The following typographical conventions are used in this book:

Constant width
Used for source code

Constant width bold

user input

[] optional elements

{ } mandatory choice

| separates choices

Summary

This whitepaper describes the scope of application level monitoring and the difference to standard server- and database monitoring. It emphasizes that application level monitoring is with very few exceptions not available as standard module of existing monitoring products and must be initiated by application support team in IS and application experts in business and not by UNIX- or database administrators.

Introduction: The three areas of monitoring

The following table shows the difference between three Monitoring Areas:

1. Hardware- and Operating System Monitoring
2. Database Monitoring
3. Application Level Monitoring

Monitoring Level	what is monitored	alarms are directed to ..	monitoring implemented using...
Hardware- and Operating System Monitoring	Monitoring of Network-card, CPU-Temperature, Fan-Status, File systems running full, alert messages in /etc/messages etc.	IT-Team	a monitoring framework, e.g. BMC-Patrol, TNG-Unicenter, HP-Openview Monitoring Pack of Oracle Enterprise Manager
Database Monitoring	Archivelog destination running full Error-Messages in alert_<sid>.log Files in user_dump_destination Tablespaces running full Table close to max. extent Limit	DBA-Team	a monitoring framework, e.g. BMC-Patrol, TNG-Unicenter, HP-Openview Monitoring Pack of Oracle Enterprise Manager
Application Level Monitoring	Application Errors Processing - Backlog Unusual (too low / too high) usage / number of processed records	Application Support	Note: for only very popular applications sold in high volume worldwide the leading vendors of monitoring frameworks sell application specific monitoring modules.

This paper focuses only on topic 3. - Application Level Monitoring.

Application-Level-Monitoring

In general "Application Level Monitoring" is NOT a discussion about purchasing a monitoring product, it's a topic of - if not part of the application - writing scripts to detect that the APPLICATION is not working. (For only very popular applications sold in high volume worldwide the leading vendors of monitoring frameworks sell application specific monitoring modules)

CMU - Common Misunderstanding in responses to a RFP (Request for Proposal)

CMU: Vendor (or internal development department) responds to the requirement for Application Level Monitoring "That's done by IT, they have a monitoring system."

Answer: No, that's WRONG, that is this common misunderstanding ! - Please read the next 2 pages.

A simple example

The customer care department complains that orders are not processed,

- IT reports that all monitoring-flags are "green", there is everything OK on the server, file systems, storage and TCP/IP-network
- DBA's report that all database monitoring flags are "green", there are no database problems

but nevertheless the orders were not processed.

That's an typical application problem beyond the scope of IT and DBA's (and beyond the standard-functionality of their monitoring modules).

Perhaps the background process processing the orders (stored for processing in an work queue (often database table) just terminated, or it's frozen (ps -ef |grep <process_name> still shows the process - but that does not mean that's working...)

or the process works fine, just problems on the interfacing system cause all orders to be put back into work queue with status "retry".

The problem in this situation is that this problem was not detected and fixed by application support team before customers complained. This missing monitoring and the resulting delayed repair of the problem could lead into violation of an SLA (Service Level Agreement) if such an SLA was agreed.

How to implement a monitor for this example ?

Approach	Type	Comment
Check for records with status "failed" and "retry" in work queue table.	error monitoring	Does not detect if processing is "frozen" (or not running at all).
Check for average number of orders processed in last time interval.	usage based monitoring	You need to collect and store some statistics about usual number of orders processed, dependent on daytime or working day and weekend
Check the log file or log table for errors	error monitoring	Does not detect if processing is "frozen" (or not running at all).
Check if logfile is growing	usage based monitoring	You need to understand what is recorded in your logfile. Just errors ? - Then it usually will never grow. Or are processed records and processing attempts ("no orders to be processed") recorded ?

Why you need application level monitoring:

.. because	comment
To ensure availability	<p>Even if you don't have an SLA (Service Level Agreement) today customers do have an expected level of availability.</p> <p>And it is quite likely that you will need to sign a SLA in near future. Service Level Management is part of ITIL (IT Infrastructure Library) which is the actual hype in IT.</p> <p>Auditing Standards (see next item) also require Service Level Management.</p>

.. because	comment
	Note: Redundant Hardware (standby-servers, clustered systems) mitigate just hardware problems causing a failover after the HA-monitor detected a problem, but it DOES NOT solve an application level problem which is not even detected.
Auditing Standards require it	The number of laws or regulations requiring external audits is increasing, e.g. SOX (Sarbanes Oxley Act) or Basel II. CobiT ® from ITGI (IT Governance Institute) and ISACA (Information Systems Auditing and Control Association) is the most famous auditing standard for IT and is used as method in those audits. To visualize the importance of auditing: CobiT dedicates one of its four domains just to Monitoring !
To meet requirements from RA (Revenue Assurance) Team	This team focuses on everything that could reduce revenue, e.g. lost orders, items not billed, .. Therefore reconciliations between systems are of special interests.

Why don't you have application level monitoring today ?

.. because	comment
we were not aware about this requirement when selecting the application or specifying requirements for development	You should use the requirements template " Non functional Requirements " from Mercury Consulting Ltd. This template with 150 non functional requirements contains a own chapter just dedicated to "Application Level Monitoring".
we asked for this requirements but non of the offered products on the markets offered those features.	Unfortunately a not unusual situation. But you should have put in your project plan the time and costs for your own or outsourced development of application level monitoring.
we plan to outsource the operations of this application, therefore we don't need it.	Be prepared: Real World Experience In one case the customer planned to outsource the operations of the application (bespoke development) after delivery, but the outsourcing company required detailed application level monitoring as mandatory before taking over the system. The development of such monitoring-scripts was ordered and delivered, resulting in significant (but unplanned) costs.

We have a monitoring framework but the vendor of the monitoring framework does not offer a module for this application

This is a very common situation. To provide application specific monitoring events requires extreme detailed understanding how the application is working, this knowledge is usually only available to the application vendor and to application support staff at customers site. But an application vendor selling his specific product 100 to 5,000 times worldwide can't afford to write monitoring modules for many existing monitoring frameworks used by his few customers.

Monitoring Frameworks usually provide "hooks" to plug in custom monitoring events: You just need a shell-script, PL/SQL-function or Java-function which can be called by the monitoring agent installed on your server and which returns a (typically numeric) return value.

In your Monitoring Framework you just define the name for this monitoring event, the script to be called and how the return-value is interpreted (e.g. return value < 5 = "OK", 5...7 = "warning", > 7 = "Alarm".

You see that the implementation is quite straight forward. Even writing the scripts is not complicated (but requires time...) -

But the important fact is that you need to know WHERE to check for WHAT!

The challenge is the DEFINITION of the application specific controls!

You can hire or contract someone with the skills to write the scripts, but the definition of the application specific controls can only be done by the application vendor or application support members, best together with Revenue Assurance (RA) team and internal auditors and the service level manager.

Other Examples

Monitored Event	Implementation		
<ul style="list-style-type: none"> for errors for backlog <p>for each order a file is stored in directory ../input and after successful processing this file is moved into directory ../archive. Files not processed because of errors remain in the ../input directory.</p>	<p>shell-script that counts the number of files in the ../input directory and raises a</p> <p>warning if exceeding <nr. of files></p> <p>alarm if exceeding <nr. of files></p> <p>warning if there is a file older than 2 minutes</p> <p>alarm if there is a file older than 5 minutes</p>		
<ul style="list-style-type: none"> for errors for backlog <p>for each order to be processed one record with status "22" is inserted into database table ORDER_WORK_QUEUE,</p> <p>status -999 indicates "in processing"</p> <p>status 50 indicates "completed"</p> <p>all values > 50 indicate "error"</p>	<p>a SQL*Statement checks for</p> <p>backlog-warning if exceeding <nr. of records "to be processed"</p> <p>backlog-alarm if exceeding <nr. of records "to be processed"</p> <p>error-warning if there are records to long in status -999 or new records in error-status.</p>		
<ul style="list-style-type: none"> for "not working" for unusual usage 	<p>SQL*statement checks for nr. of orders processed in last 5 minutes and raises warnings and alarms based on following table:</p>		
	Time	warning	alarm
	22:00-1:59am	last 30 minutes < 5	last 30 minutes = 0
	2:00am-4:59am	no warning	last 2 hours = 0
	5:00am-6:59am	last 30 minutes < 5	last 30 minutes = 0
	7:00am-19:59	last 5 minutes < 5	last 5 minutes = 0
	20:00-21:59	last 5 minutes = 0	last 15 minutes = 0
<ul style="list-style-type: none"> for unusual CPU-consumption <p>certain application processes are known to consume even under full load (e.g. backlog in work queue) never more than 20% of one CPU, and it is known that if this process shows 100% CPU consumption the process was looping in an error-status.</p>	<p>simple shell-script that greps the output of "top" for processes consuming hight CPU and matching with list of processes which should not consume that much CPU.</p> <p>In case that shell script detects a suspect process full on CPU it raises an alarm.</p>		

How Mercury Consulting Ltd. can help you

Mercury Consulting Limited (MCL) is a professional consultancy providing experience, support and training in IS/IT operations for companies during high-time-pressure startup phase and following consolidation phase, especially in the Telecom-market.

Our Products

you can purchase online and immediately download at our eBook-Shop <http://www.IT-Checklists.com>



Our Checklists and Templates will help you to ensure that good or best practice is not only known but consistently applied!

Database Independent Products

Product	Benefit
150 Non functional Requirements.	Requirements Template with 150 non functional requirements for selecting or developing robust, reliable, stable and manageable applications to meet the Service Level Agreements (SLA's). For external RFP's (Request for Proposal) and for internal development.
Checklist for Data Migration	65 important questions to identify and address or exclude typical migration pitfalls in an early phase of the project, thus ensuring the confidence for keeping the time plan.
Template: Systems- and Operations Manual	Template to establish that documentation auditors like to see!
Interface Checklist	Those questions which you need to ask before starting the development! Requirements, Checklist and Template for Planning, Defining and Documenting Application Interfaces.
Checklist for Production Release and System Handover	Checklist for small and medium projects focusing on non-functional aspects for operations team. Simple but effective!
Business Requirements for Archiving and Purging	Template with Business Requirements for Archiving & Restore & Purging. Not removing old customer data can cause conflicts with privacy laws. Business must act and clearly specify what to purge and what to archive!
Application Health Check: Stability Assessment	Using this template to check your systems - and DOCUMENT the findings might show you even more potential issues beyond invalid objects.
Technology Selection for Disaster Recovery (DR)	Standby database using transfer of transaction log / archive logs or replicating the database files ? Host-based, Storage based or Switch/Fabric based replication ? This document evaluates the different technologies and their advantages and constraints from an operational point of view.
Backup SLA / OLA: Operations Level Agreement with the Backup-Team	A template which supports the creation of an operational level agreement (OLA) between application support, database administrators and backup team.

Database Specific Products

Product	Benefit
Template for Database Operations Level Agreement (OLA) / SLA	<p>If a Service Level Manager needs to offer to Business a Service Level Agreement (SLA) for an End-to-End IS/IT-Service, he sign this SLA only after he arranged within IS/IT for each system or component used to provide this service an Operations Level Agreement (OLA) with the providing team or department.</p> <p>This document provides a template for such an Operations Level Agreement (OLA) for Databases containing the agreed values and for QA-purpose also the measures implemented to reduce the likelihood of violations of those agreed values.</p> <p>It does not only deal with availability, but contains also comprehensive service catalogue of advanced DBA services and a template for the Service Level Reporting (SLR).</p>
Database Health Check - Part 1: Stability Assessment	<p>Stability Assessment of your Database.</p> <p>Most Application- and Database-Crashes can be avoided when detecting early indicators and reacting to them.</p> <p>Be Proactive - Check Now!</p>
DBA and Application Support: Job Description and Self Assessment	<p>Checklist to ensure that all 60 DBA-duties are assigned:</p> <p>System DBA, Development DBA and Application DBA versus Application Support</p> <p>If you have just a job role "DBA", but not a dedicated job role "Application DBA" those 19 duties must be explicitly assigned to either "Application Support" or to the "DBA" - otherwise they might not be executed!</p> <p>Detect unassigned tasks before an auditor reports them !</p> <p>This product addresses disputes between System DBA and Application Support (or, if existing, dedicated Application DBA) regarding the responsibility for the application's database objects.</p>

Our free Whitepapers

<http://www.mercury-consulting-ltd.com/wp/whitepapers.html>

Whitepaper	Benefit and Description
The Importance of Application Level Monitoring.	Keeping your applications free from invalid objects is an important task, but does not guarantee error free operations. This free whitepaper explains the difference between "Application Level Monitoring" to "Database Monitoring" and "System/Server/OS"-Monitoring.
Important Facts about Redolog- and Archivelog / Transaction Logs in Databases for Change Managers and Application Support Staff	Understanding the topic "archivelog volume" can avoid unexpected troubles when applying changes or conducting upgrades. This awareness paper explains why Change Managers must ask questions about the archivelog volume created during changes and upgrades and why application support staff – and not only the DBA - must understand this topic.
The Danger of Invalid Database Objects	An awareness paper for Operations Managers and Application Support describing the problems and potential risks caused by invalid objects in Databases.
Private or Public Synonyms – or no Synonyms at all ?	A decision support paper visualizing the pro's and contras on a single page in tabular form and evaluating the arguments.
Daily, weekly or monthly partitions?	This document evaluates the different factors impacting the decision if daily, weekly or monthly partitions should be used. Legal- , performance- and maintenance aspects are evaluated.

Our Services

<http://www.Mercury-Consulting-Ltd.com/services.htm>

About the Author

The Author of this white bears following credentials



and has 12 years experience as DBA and 6 years experience in Telecommunications Companies.