

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Advanced Oracle monitoring agent for Zabbix

Francisco Barros

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

FEUP Supervisor: Gabriel David

November 16, 2022

Resumo

A monitorização de sistemas IT é uma ferramenta muito importante para as empresas dos dias de hoje, que necessitam de garantir que todos os seus sistemas estejam sempre ao nível esperado. Sendo as bases de dados um componente essencial para a operação destas empresas, este projeto tem como objetivo demonstrar a utilização de uma das mais conhecidas ferramentas de monitorização, o Zabbix, para a monitorização de bases de dados Oracle. Todos os resultados obtidos e todos os testes realizados foram efetuados com a utilização de uma base de dados genérica de testes. Os resultados demonstram como será possível obter um claro aumento da eficiência na gestão de bases de dados e o quão mais seguros os sistemas e as bases de dados se tornam com a atualização permanente dos itens monitorizados e com os alertas automáticos.

Abstract

The monitoring of IT systems is a very important tool for today's companies, which need to ensure that all their systems are always at the expected level. Being databases an essential component for the operation of these companies, this project aims to demonstrate the use of one of the most well known monitoring tools, Zabbix, for monitoring Oracle databases. All the results obtained and all the tests performed were done using a generic test database. The results show how it is possible to obtain a clear increase of efficiency in database management and how much more secure the systems and databases become with the permanent update of the monitored items and automatic alerts.

Agradecimentos

Gostaria de agradecer esta dissertação à minha família e namorada pelo apoio que sempre me deram durante a realização desta tese. Quero também agradecer ao Jorge Duarte da empresa IT Peers pela oportunidade que me deu para poder integrar neste projecto e pela ajuda que sempre prestou. Por fim, quero deixar um especial agradecimento ao meu orientador Gabriel David pela incansável ajuda e por se mostrar sempre disponível para prestar qualquer ajuda.

Francisco Barros

Contents

1	Introduction	1
2	IT Systems Monitoring	3
2.1	Generic Monitoring Architecture	3
2.2	Basic Types of IT Monitoring	4
2.3	Database Monitoring	5
2.3.1	Oracle Database Systems	6
2.3.2	Oracle System Views	7
2.3.3	Oracle Enterprise Manager	8
2.4	Zabbix	9
2.4.1	Features	9
2.4.2	Architecture	11
2.4.3	Agent 2	13
2.4.4	Plugins	13
2.5	Relational Database	14
3	Plugin to monitor Oracle databases	17
3.1	Installation	17
3.1.1	Oracle Instant Client installation	17
3.1.2	Oracle DB user permissions	20
3.2	Configuration	20
3.2.1	Connection Configuration	21
3.3	Custom Queries	22
3.4	Supported Keys	22
3.5	Oracle by Zabbix agent 2	24
3.5.1	Macros	25
3.5.2	Triggers	27
3.5.3	Items	34
4	Obtained Results	43
5	Conclusion	49
5.1	Future Work	49
	References	53

List of Figures

2.1	Infrastructure monitoring architecture [1]	4
2.2	Single Instance vs. RAC [2]	7
2.3	Active Checks [3]	14
2.4	Passive Checks [3]	15
4.1	Items collected 1/5	43
4.2	Items collected 2/5	44
4.3	Items collected 3/5	44
4.4	Items collected 4/5	44
4.5	Items collected 5/5	45
4.6	Graph about the Datafiles	45
4.7	Graph about the Processes	45
4.8	Graph about the Sessions	45
4.9	Graph about the PGA	46
4.10	Graph about the SGA	46
4.11	Graph about the Physical Read Write Rate	46
4.12	Graph about the Physical Read Write Rate in bytes	46
4.13	Graph about the Tablespace 'SYSAUX'	47
4.14	Zabbix dashboard, including alerts raised by triggers	47

List of Tables

3.1	Information collected about the database application	37
3.2	Information collected about SGA	37
3.3	Information collected about PGA	38
3.4	Information collected about FRA	38
3.5	Information collected about the #TABLESPACES	39
3.6	Information collected about the #DBNAME	39
3.7	Information collected about the sessions	40
3.8	Cache-related Items	41
3.9	Information related to the CPU	41
3.10	Information related to the Disk	41
3.11	Information related to the System	41
3.12	Information related to the Datafiles	41
3.13	Raw Items collected	42

Abbreviations and Symbols

ADT	Abstract Data Type
ANDF	Architecture-Neutral Distribution Format
API	Application Programming Interface
APM	Application Performance Management
ASM	Automatic Storage Management
BAM	Business Activity Monitoring
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CDB	Container Database
CORBA	Common Object Request Broker Architecture
DBA	Database Administrator
DBWR	Database Write
DBMS	Database Management Systems
FRA	Fast Recovery Area
GI	Grid Infrastructure
GV\$	Global Dynamic Performance V\$
IPMI	Intelligent Platform Management Interface
IT	Information technology
JDBC	Java Database Connectivity
JMX	Java Management Extensions
LLD	Low-Level Design
OCCI	Oracle C++ Call Interface
OCI	Oracle Call Interface
ODBC	Open Database Connectivity
OID	Object Identifier
ORCL	Oracle
PDB	Pluggable Database
PGA	Program Global Area
PHP	Hypertext Preprocessor (Personal Home Page)
RAC	Real Application Cluster
RDBMS	Relational Data Base Management System
RPM	Red Hat Package Manager
RUM	Real user monitoring
SDK	Software Development Kit
SGA	System Global Area
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UNCOL	UNiversal COmpiler-oriented Language
URI	Uniform Resource Identifier
V\$	Dynamic Performance Views
WWW	<i>World Wide Web</i>

Chapter 1

Introduction

The world is becoming more and more automated. Systems are getting more advanced and complex and human hand is becoming less and less needed. Companies now have a vast amount of machines running servers, programs and complex databases. But how can we assure that these systems, machines and databases stay healthy and running properly? Imagine having people running tests on various machines one by one to check for problems. That's when the need for monitoring platforms appears. These platforms perform checks periodically and issue alerts to the company managers when a problem appears. These problems can be solved automatically by the platform or, when a more complex problem occurs, some worker can go and have a look. This saves hours and hours of problem checking and let the workers spend their time on different tasks.

Oracle databases have a poor integration with Zabbix even though the capabilities provided are good and so, this paper will show how to properly install and configure the Zabbix Oracle template and a demonstration of its use.

The document is organized as follows. After the introductory first chapter, chapter 2 provides background knowledge about IT monitoring and Oracle systems and presents a review of the state-of-the-art. It is followed by chapter 3 which talks about the Oracle plugin on the Zabbix platform. Chapter 4 comes with the results obtained after running some tests. The document then finishes with a conclusion chapter and a brief text about future work.

Chapter 2

IT Systems Monitoring

In the early days of IT, the systems were simple compared to today's standards. IT teams often consisted of people who could fix problems that occurred just because there weren't too many unique. [4]

As we started to evolve, we gradually needed to allow for specialization. The individual systems themselves became so complex that domain-specific specialists were required to monitor, maintain, and manage each system. [4]

Furthermore, today's systems are sometimes geographically distributed or run on various operating systems, making management more difficult. This issue has reignited interest in an old topic: IT infrastructure and application monitoring. [1]

Also, as more businesses rely on software, the health of IT systems is crucial for providing 24/7 customer service and support. The systems can be on-premises applications, cloud-based apps, or any combination of the two. [1]

2.1 Generic Monitoring Architecture

IT monitoring is a broad category of tools that allow analysts to verify whether IT equipment is online and working at expected levels while also correcting any issues that are discovered. IT monitoring tools range from basic inspections to more powerful tools that can granularly assess a product's performance or even automate repairs when problems are identified.

When used with a web server, an IT monitoring tool can look at patterns in CPU and RAM usage, see if hard drive space is running out, and see how often the system has crashed in the last year.

And, with hundreds of monitoring products and solutions on the market, monitoring systems varies significantly in terms of complexity and strategy. As a result, one company's ideal IT monitoring tool or monitoring software may be a poor fit for another.

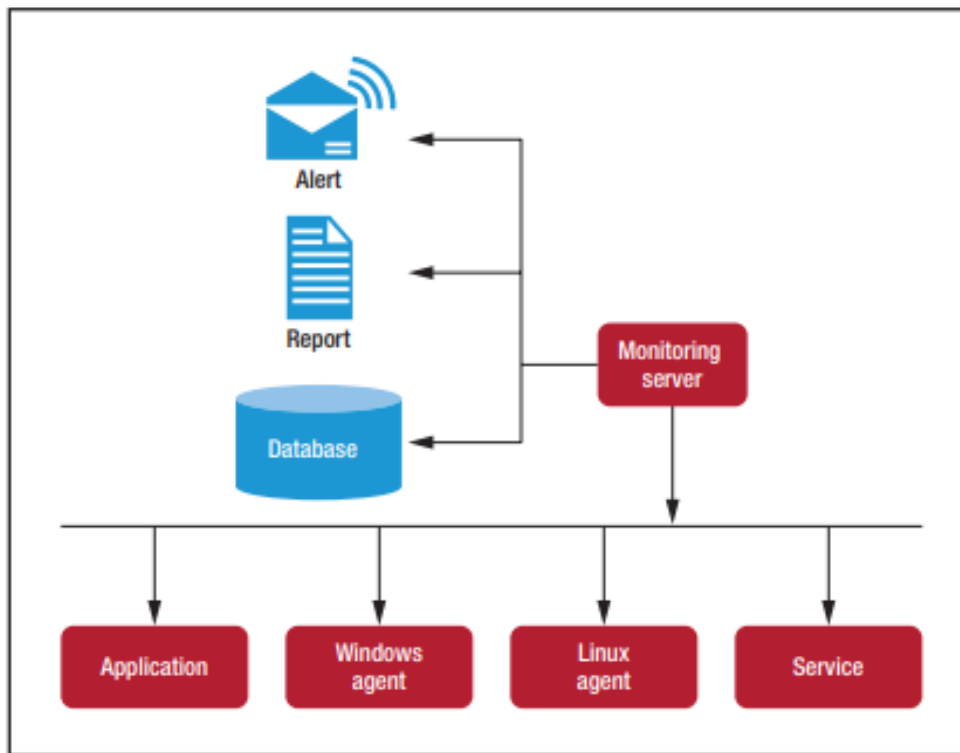


Figure 2.1: Infrastructure monitoring architecture [1]

2.2 Basic Types of IT Monitoring

The basic types of IT monitoring include availability monitoring, web performance monitoring, web application management and application performance management, API management, real user monitoring, security monitoring and business activity monitoring. [5]

While there isn't a definitive list of the numerous types of IT monitoring tools, many phrases encompass multiple monitoring types, blurring the distinctions that characterize this sector. Here are some of the most common types of IT monitoring tools: [5]

- **Availability monitoring:** Often known as system monitoring, is the most advanced sort of IT monitoring tool. This covers categories including server management, infrastructure monitoring/management, and network monitoring/management, all of which are aimed to give customers information regarding uptime and performance of whatever is being monitored. [5]
- **Web performance monitoring:** A subset of availability monitoring that is aimed to monitor the availability of a web server or service while also adding more fine-grained detail to the system. These tools can gather data like page loading time, error location, and individual load times of various web components, allowing analysts to smooth the performance of a website or web-based app. [5]

- **Application management/Application Performance Management (APM):** APM tools are comparable to web performance monitoring tools, but they're specifically developed for customer-facing apps, allowing analysts to follow an application's performance and identify any issues before they become too serious for the user base. Modern APM solutions can feature automatic algorithms that can troubleshoot these issues without the need for a human developer's interaction. [5]
- **API monitoring:** API monitoring tools and software provide insight into whether or not an API is functioning effectively, ensuring that downtime is kept to a minimum. [5]
- **Real user monitoring (RUM):** Actual end-user interactions with a website or application are recorded using real user monitoring. It can discover problems based on "actual" user experience challenges by monitoring real-world load times and user behavior. This sort of monitoring is intended to be retrospective rather than predictive, allowing analysts to detect issues only after they have occurred. [5]
- **Security monitoring:** Designed to look for breaches or other unexpected behaviour on a network. Security monitoring is a broad area that encompasses a variety of subsets of security analysis technologies. [5]
- **Business activity monitoring (BAM):** Keeps track of critical business performance measures throughout time. These metrics could include data on retail sales, app downloads, or the amount of financial transfers, for example. [5]

2.3 Database Monitoring

The systems that have been chosen as targets for monitoring experiences in this dissertation are database systems and, more specifically, Oracle database systems.

The task of assessing the operating status of a database is referred to as database monitoring. [6]

Database monitoring is essential for keeping a database management system's performance and health in good shape. The database management system collects data from the database manager, its databases, and any associated applications to make monitoring easier. One can conduct the following tasks and more with this information: [6]

- Forecast hardware requirements based on database usage patterns.
- Examine how well specific applications or SQL queries execute.
- Keep track of how often indexes and tables are used.
- Determine the source of the system's poor performance.
- Analyze the impact of optimization efforts.

2.3.1 Oracle Database Systems

Oracle database products offer customers cost-optimized and high-performance versions of Oracle Database, the world's leading converged, multi-model database management system, as well as in-memory, NoSQL and MySQL databases. Oracle Autonomous Database enables customers to simplify relational database environments and reduce management workloads [7].

There are considerable advantages in using Oracle Databases:

- **Reduced operational costs** - Reduce the costs of managing, securing, and maintaining your Oracle databases with machine learning-driven automation. Provision, scale and tune, protect and patch, and repair the database—all automatically without user intervention [7].
- **Guard against data breaches** - Assess, detect, and prevent data security threats with Oracle database security solutions for encryption, key management, data masking, privileged user access controls, activity monitoring, and auditing. Reduce the risk of a data breach and simplify and accelerate compliance [7].
- **Use of a single database for all data types** - Free application developers from complex transformations and redundant data with Oracle's converged database [7].
- **Flexibility on deployment** - Deploy Oracle Database wherever required (data center, public cloud, or private cloud). This offers the flexibility between deployment in your data center when residency or latency are critical, or in the cloud when you want to take advantage of scalability and the broadest set of capabilities [7].

Oracle Database may be set up in two ways: as a single instance (non-cluster) database or as a Real Application Cluster (RAC) database. The instance (memory and processes) and the database have a one-to-one relationship in a single instance database (database files) and the Oracle software, database files, memory and processes are all housed on a single server. The database and instances in Oracle RAC settings, on the other hand, have a one-to-many relationship. An Oracle RAC database can contain up to 100 instances that all connect to the same database. All the servers for a single database in the Oracle RAC environment must belong to the same Oracle Real Application Cluster. [2]

Each Real Application Cluster has Grid Infrastructure (GI) installed locally on each node, which is something that all Real Application Clusters share in common. Grid Infrastructure includes several products: Automatic Storage Management, Oracle Clusterware, and others. [2]

The database files on shared storage are accessed by each instance in the Oracle RAC cluster. In addition, each node requires local storage for the GI and Oracle Database software. [2]

A cluster comprises multiple interconnected computers or servers that appear as if they are one server to end users and applications. Oracle Database's RAC feature allows one to cluster Oracle databases. Oracle RAC relies on Oracle Clusterware to connect several servers and make them work as a single system. [2]

Oracle Clusterware is an Oracle Database-integrated portable cluster management tool. Oracle Clusterware is a prerequisite for running Oracle RAC since it offers the infrastructure required to

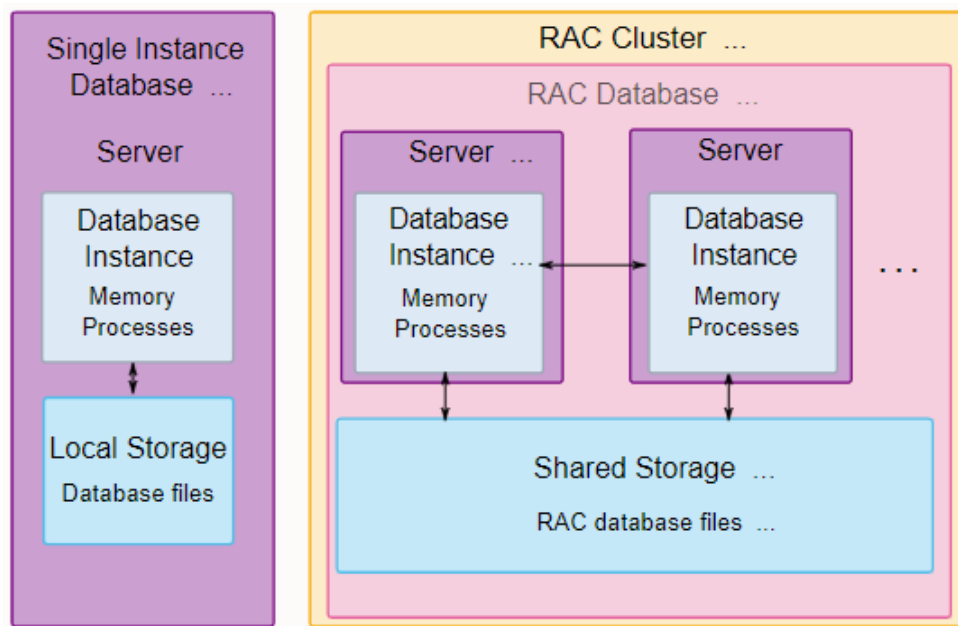


Figure 2.2: Single Instance vs. RAC [2]

run Oracle RAC. Oracle Clusterware also handles resources including Virtual Internet Protocol (VIP) addresses, databases, listeners, and services, among other things. Oracle Clusterware also allows non-cluster Oracle databases as well as Oracle RAC databases to utilise Oracle's high-availability infrastructure. Oracle Clusterware, in conjunction with Oracle Automatic Storage Management (Oracle ASM), enables you to establish a clustered storage pool that may be used by any combination of noncluster and Oracle RAC databases. [2]

Oracle Clusterware is the only clusterware you'll ever need for most Oracle RAC platforms. If your database applications demand vendor clusterware, you can utilize it in conjunction with Oracle Clusterware if the vendor clusterware is Oracle RAC approved. [2]

2.3.2 Oracle System Views

Views are customized presentations of data in one or more tables or other views. They can also be seen as stored queries. These views don't contain any data, but instead obtain their data from the tables they are based on. These tables are referred to as the base tables of the view.

Because a result of query is conceptually a table, a view can be used like a table in further queries. By restricting access to a predetermined group of rows and columns in a table, views can add an extra layer of protection. They can also hold sophisticated queries and hide data complexity.

In a database, besides the user tables and other database objects, there are system tables and views that are collectively called dictionary, because they contain information on the database operation. There are two types: static data dictionary views and dynamic performance views.

Static Views are the data dictionary views since they only change when a change occurs to the data dictionary. Examples of data dictionary changes include creating a new table or granting a

privilege to a user.

Almost all data dictionary tables have three corresponding views:

- A `DBA_` view displays all relevant information in the entire database. These are only for administrators. For example, the view `DBA_TABLES` shows one row for each table in the database.
- An `ALL_` view displays all the information accessible to the current user, including information from the schema of the current user, and information from objects in other schemas, if the current user has access to those objects through privileges or roles. For example, the view `ALL_TABLES` shows one row for each table accessible to the current user.
- A `USER_` view displays all the information from the schema of the current user. For example, the view `USER_TABLES` shows one row for each table in the current user schema

The columns in the `DBA_`, `ALL_`, and `USER_` views are usually nearly identical. The `USER_` view usually does not have an `OWNER` column.

Dynamic Performance Views keep track of what's going on in the database. Only admins have access to these.

Oracle Enterprise Manager, which is the principal interface for accessing information regarding system performance, uses dynamic performance views. The `V$` views that read from memory are accessible once an instance is started. Some views that read data from disk require the database to be mounted, while others require the database to be open.

`V$` views are `CONTAINER_DATA` objects. When a user connected to the root queries a `V$` view, the query results will depend on the `CONTAINER_DATA` attribute for users for the view. The `CONTAINER_DATA` clause of the `SQL ALTER USER` statement is used to set and modify users' `CONTAINER_DATA` attribute.

An example of a `V$` view is `V$SGA`, which returns the current sizes of various System Global Area (SGA) memory components.

Oracle has a `GV$` (global `V$`) view that corresponds to practically every `V$` view. Oracle Real Application Clusters make use of these views (RACs). Querying a `GV$` view returns information about the `V$` view from all qualifying instances. Each `GV$` view has an extra field entitled `INST_ID` of data type `NUMBER` in addition to the `V$` information. The `INST_ID` field shows the instance number from which the `V$` view information was retrieved. `V$` information can be retrieved from a subset of available instances using the `INST_ID` column as a filter.

2.3.3 Oracle Enterprise Manager

The majority of DBAs (Database Administrators) spend far too much time on routine maintenance and troubleshooting. As a result, they have less time to devise solutions to improve the important business systems within their control. In response, the IT industry has begun to make a dramatic shift to cloud services in order to free up experienced technological personnel for more

productive endeavors. Most businesses will have a hybrid estate as a result of this change, which will necessitate new management tools and practices. Today's businesses want flexible technologies to help them streamline the process of transferring resources to the cloud and manage hybrid environments. [8].

Oracle Enterprise Manager is an on-premise management tool that lets you manage all of your Oracle deployments, whether they're in the data center or in the cloud, from a single dashboard. It delivers market-leading management and automation support for Oracle applications, databases, middleware, hardware, and engineered systems through deep integration with Oracle's product stack. [8]. Customers depend on Oracle Enterprise Manager because it delivers:

- Increased visibility and intelligent analytics
- Comprehensive life cycle automation and control
- An enterprise-grade management platform that is secure, accessible, and extensible

2.4 Zabbix

Zabbix is a network monitoring tool developed and sponsored by Zabbix SIA, which was founded in 1998 by Alexei Vladishev.

It's an enterprise-class open source distributed monitoring tool that can track a network's health and integrity, as well as the health and integrity of servers, virtual machines, applications, services, databases, websites, and the cloud. Zabbix has a versatile notification system that lets users to up e-mail-based notifications for almost any occurrence. This enables a quick response to server issues. Based on the recorded data, Zabbix provides strong reporting and data visualization options. Zabbix is therefore suitable for capacity planning. [9].

2.4.1 Features

Data gathering:

- Availability and performance checks
- Support for SNMP (both trapping and polling), IPMI, JMX, VMware monitoring
- custom checks
- gathering desired data at custom intervals
- performed by server/proxy and by agents

Flexible threshold definitions:

- you can define very flexible problem thresholds, called triggers, referencing values from the back-end database

Highly configurable alerting:

- sending notifications can be customized for the escalation schedule, recipient, media type
- notifications can be made meaningful and helpful using macro variables
- automatic actions include remote commands

Real-time graphing:

- monitored items are immediately graphed using the built-in graphing functionality

Web monitoring capabilities

- Zabbix can follow a path of simulated mouse clicks on a web site and check for functionality and response time

Extensive visualization options:

- ability to create custom graphs that can combine multiple items into a single view
- network maps
- slideshows in a dashboard-style overview
- reports
- high-level (business) view of monitored resources

Historical data storage:

- data stored in a database
- configurable history
- built-in housekeeping procedure

Easy configuration:

- add monitored devices as hosts
- hosts are picked up for monitoring, once in the database
- apply templates to monitored devices

Use of templates:

- grouping checks in templates
- templates can inherit other templates

Network discovery:

- automatic discovery of network devices
- agent auto-registration 7
- discovery of file systems, network interfaces and SNMP OIDs

Fast web interface:

- a web-based front-end in PHP
- accessible from anywhere
- you can click your way through
- audit log

Zabbix API:

- Zabbix API provides programmable interface to Zabbix for mass manipulations, 3rd party software integration and other purposes.

Permissions system:

- secure user authentication
- certain users can be limited to certain views

Full featured and easily extensible agent:

- deployed on monitoring targets
- can be deployed on both Linux and Windows

Binary daemons:

- written in C, for performance and small memory footprint
- easily portable

Ready for complex environments:

- remote monitoring made easy by using a Zabbix proxy

2.4.2 Architecture

Zabbix consists of several major software components, the responsibilities of which are outlined below.

- **Server** - Zabbix server is the central component to which agents report availability and integrity information and statistics. The server is the central repository in which all configuration, statistical and operational data are stored [9].

- **Database storage** - All configuration information as well as the data gathered by Zabbix is stored in a database [9].
- **Web interface** - For an easy access to Zabbix from anywhere and from any platform, the web-based interface is provided. The interface is part of Zabbix server, and usually runs on the same physical machine as the one running the server [9].
- **Proxy** - Zabbix proxy can collect performance and availability data on behalf of Zabbix server. A proxy is an optional part of Zabbix deployment; however, it may be very beneficial to distribute the load of a single Zabbix server [9].
- **Agent** - Zabbix agents are deployed on monitoring targets to actively monitor local resources and applications. The agent gathers operational information locally and reports data to Zabbix server for further processing. In case of failures (such as a hard disk running full or a crashed service process), Zabbix server can actively alert the administrators of the particular machine that reported the failure [9].

Zabbix agents can perform passive and active checks.

In a passive check the agent responds to a data request. Zabbix server (or proxy) asks for data, for example, CPU load, and Zabbix agent sends back the result [10].

Active checks require more complex processing. The agent must first retrieve a list of items from Zabbix server for independent processing. Then it will periodically send new values to the server [10].

- **Agent 2** - Zabbix agent 2 is a new generation of Zabbix agent and may be used in place of Zabbix agent. Zabbix agent 2 has been developed to:
 - reduce the number of TCP connections
 - provide improved concurrency of checks
 - be easily extendable with plugins. A plugin should be able to:
 - * provide trivial checks consisting of only a few simple lines of code
 - * provide complex checks consisting of long-running scripts and standalone data gathering with periodic sending back of the data
 - be a drop-in replacement for Zabbix agent (in that it supports all the previous functionality)

Agent 2 is written in Go (with some C code of Zabbix agent reused).

Passive checks work similarly to Zabbix agent. Active checks support scheduled/flexible intervals and check concurrency within one active server [10].

- **Templates** - A template is a set of entities that can be conveniently applied to multiple hosts. The entities may be:
 - items

- dashboards
- triggers
- graphs
- low-level discovery rules
- web scenarios

Because many hosts are identical or very similar in real life, the set of entities (items, triggers, graphs, etc.) developed for one host may be beneficial for many others. Those entities could, of course, be copied to each new host manually, but it would be a lot of work. Instead, with templates, they can be combined into a single template, and then all that is required is to apply the template to hosts [10].

Another advantage of using templates is that they can be used when anything needs to be altered for all hosts. If you make a change to the template, it will be replicated across all associated hosts. [10].

As a result, using templates is a great method to reduce your burden and streamline your Zabbix settings [10].

2.4.3 Agent 2

Since this dissertation is focused around the development of a plugin for zabbix, agent 2 will be the only agent analyzed in more detail since it is the only one that supports plugins.

The three main components of the agent are **ServerConnector**, **ServerListener**, and **Scheduler**.

ServerConnector manages communication with the server (receiving configuration/exporting data), item configuration, and history cache. There is one **Connector** per active server [3].

ServerListener accepts passive check requests from the server and then sends them to **Scheduler** [3].

Scheduler manages the task queue according to the schedule and concurrency settings. The Agent 2 launches a single **Scheduler** to manage tasks (plugins) in accordance with the schedule defined by item settings [3].

We can look at the inner structure of the agent in terms of active and passive checks as mentioned in the subsection 2.4.2. Even though they share the same components, they work differently from each other. To better demonstrate how they work, the following schemes (Figures 2.3 and 2.4) illustrate components interactions for each type.

2.4.4 Plugins

Plugins provide an option to extend the monitoring capabilities of Zabbix.

Plugins are a replacement for loadable modules (written in C) and other techniques for extending Zabbix capabilities, such as user parameters (agent metrics), external checks (agent-less monitoring), and `system.run[] Zabbix agent item` [9].

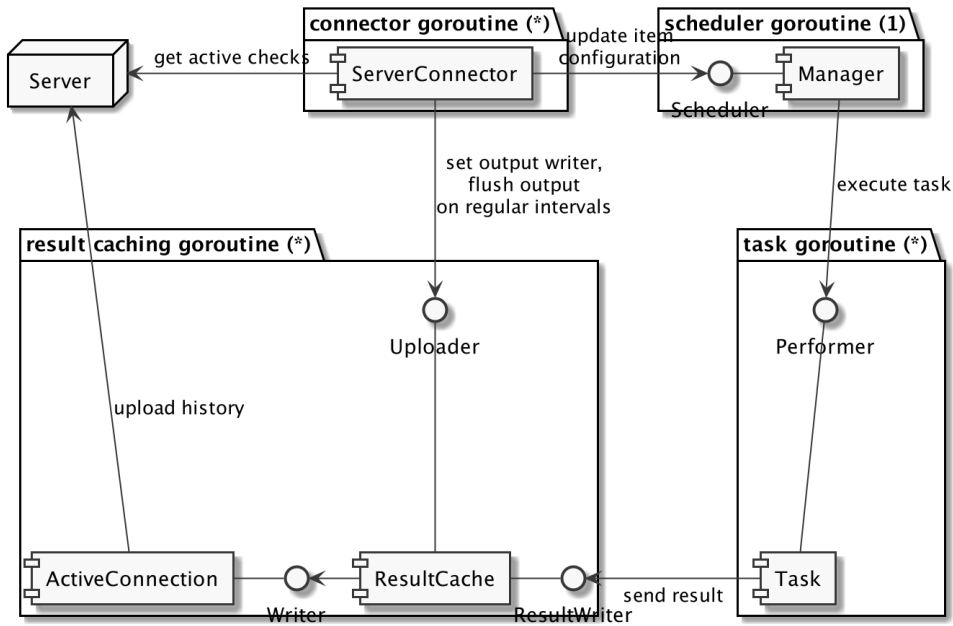


Figure 2.3: Active Checks [3]

2.5 Relational Database

A relational database is a type of database that stores and provides access to data points that are related to one another [11].

All data in relational databases is organized in tables with columns and rows, with each row containing a unique ID called the primary key and each column containing the data's properties. The foreign key notion is used to connect tables.

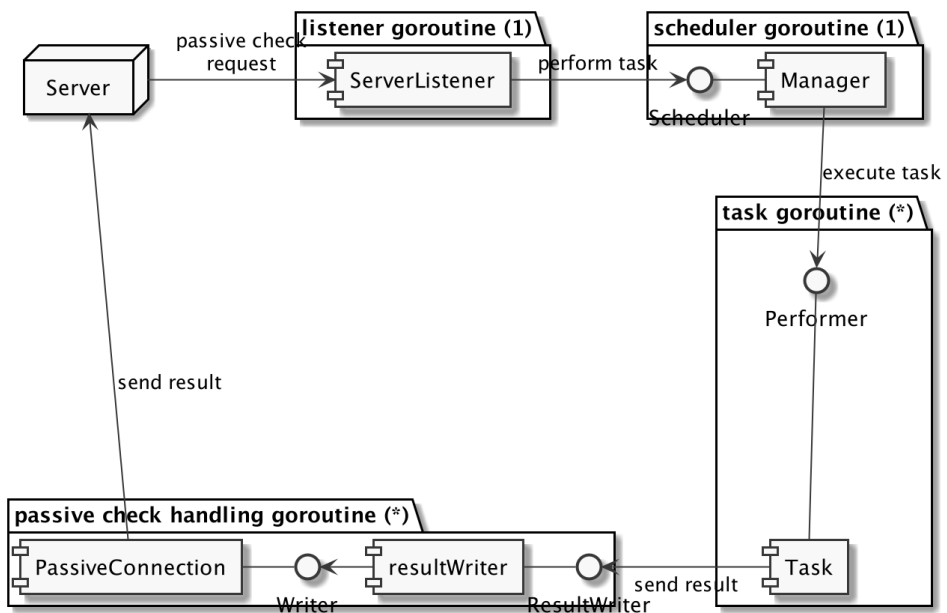


Figure 2.4: Passive Checks [3]

Chapter 3

Plugin to monitor Oracle databases

Zabbix's Oracle Database Plugin offers a native Zabbix monitoring solution for Oracle Database (multi-model database management system). Multiple Oracle instances, whether local or remote to the Zabbix Agent, can be monitored at once. To lessen network traffic, latency, CPU, and memory utilization, the plugin keeps connections open. The use of the standard Oracle template (section 3.5) is highly recommended. In this dissertation, the template is extended and modified to allow for new functionalities.

3.1 Installation

3.1.1 Oracle Instant Client installation

To start with, Zabbix's Oracle Database Plugin works with the Oracle instant client. Applications that connect to Oracle Database can be created and deployed using Oracle Instant Client, whether the database is on-premise or in the cloud. To fully utilize Oracle Database, the Instant Client libraries offer the network connectivity and cutting-edge data features required. The libraries provide access for Oracle Call Interface (OCI), Oracle C++ Call Interface (OCCI), JDBC OCI, ODBC, and Pro*C applications in addition to being used by the Oracle APIs of well-known programming languages and environments like Python, Node.js, Go, PHP, and Ruby. Quick and convenient data access is made possible by tools like SQL*Plus, SQL*Loader, and Oracle Data Pump that are part of Instant Client.[12]

In this dissertation the installation was made on a Linux Ubuntu system. The followed steps can be following for the installation.

Install RPMs

1. Download the Oracle Instant Client RPM files from oracle. All users require "Basic" or "Basic lite," and the majority of users desire "SQL*Plus" and the "SDK." [13]
2. Install the prerequisites. [13]

```
sudo apt-get install libaiol
```

3. In Ubuntu, these .rpm files need to be converted into .deb packages and installed using the "alien" command.[\[13\]](#)
4. Using version 12.1.0.2.0-1 for Linux x86 64 (64-bit) as an example:

```
alien -i oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
alien -i
  ↪ oracle-instantclient12.1-sqlplus-12.1.0.2.0-1.x86_64.rpm
alien -i oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
```

5. The installation can be tested by using "sqlplus" or "sqlplus64" to connect to your database:[\[13\]](#)

```
sqlplus username/password@//dbhost:1521/SID
```

Integrate Oracle Libraries

If Oracle applications, such as sqlplus, are complaining about missing libraries, you can add the Oracle libraries to the LD_LIBRARY_PATH each time it is used:[\[13\]](#)

```
export LD_LIBRARY_PATH=/usr/lib/oracle/<version>/client(64)/lib|
  ↪ /${LD_LIBRARY_PATH:+:$LD_LIBRARY_PATH}
```

Using 12.1 version for Linux x86_64 as an example:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/${LD_L|
  ↪ IBRARY_PATH:+:$LD_LIBRARY_PATH}
```

or to add it to the system library list create a new file as follows:

```
sudo vi /etc/ld.so.conf.d/oracle.conf && sudo chmod o+r
  ↪ /etc/ld.so.conf.d/oracle.conf
```

and add the Oracle library path as the first line. For example,

```
/usr/lib/oracle/12.1/client64/lib/
```

or

```
/usr/lib/oracle/11.2/client/lib/
```

Then run ldconfig:

```
sudo ldconfig
```

ORACLE_HOME

The location indicated by the environment variable "ORACLE HOME" is where many Oracle database applications look for Oracle software. Typical workstations will only have one Oracle installation, therefore you should declare this variable somewhere that is accessible to the entire system.[13]

```
sudo sudo vi /etc/profile.d/oracle.sh && sudo chmod o+r
→ /etc/profile.d/oracle.sh
```

Add the following:

```
export ORACLE_HOME=/usr/lib/oracle/<version>/client(64)
```

For example:

```
export ORACLE_HOME=/usr/lib/oracle/12.1/client64
```

Alternatively, each user can define this in their `/.bash_profile`

SDK fix

Some packages may check `$ORACLE_HOME/include` or `$ORACLE_HOME/rdbms/public` for the file "oci.h"[13]

`/usr/include/oracle/<version>/client` is where the instant client occasionally stores include files like `oci.h`.[13]

Inspect your system by running the following commands:

```
ls \${ORACLE_HOME}
ls -d /usr/include/oracle/*/client*/*
```

If there is no 'include' directory under `ORACLE_HOME`, and it is located over in `/usr/include/oracle/`, create a symbolic link to assist packages looking for these header files. For example,

```
sudo ln -s /usr/include/oracle/11.2/client \${ORACLE_HOME}/include
```

or:

```
sudo ln -s /usr/include/oracle/12.1/client64 \${ORACLE_HOME}/include
```

And then check if it is correct

```
ls \${ORACLE_HOME}
```

3.1.2 Oracle DB user permissions

After the installation of the Oracle Instant Client, a Oracle DB user should be created and granted permissions:

```
CREATE USER zabbix_example IDENTIFIED BY <PASSWORD>;
-- Grant access to the zabbix_example user.
GRANT CONNECT, CREATE SESSION TO zabbix_example;
GRANT SELECT ON DBA_TABLESPACE_USAGE_METRICS TO zabbix_example;
GRANT SELECT ON DBA_TABLESPACES TO zabbix_example;
GRANT SELECT ON DBA_USERS TO zabbix_example;
GRANT SELECT ON SYS.DBA_DATA_FILES TO zabbix_example;
GRANT SELECT ON V$ACTIVE_SESSION_HISTORY TO zabbix_example;
GRANT SELECT ON V$ARCHIVE_DEST TO zabbix_example;
GRANT SELECT ON V$ASM_DISKGROUP TO zabbix_example;
GRANT SELECT ON V$DATABASE TO zabbix_example;
GRANT SELECT ON V$DATAFILE TO zabbix_example;
GRANT SELECT ON V$INSTANCE TO zabbix_example;
GRANT SELECT ON V$LOG TO zabbix_example;
GRANT SELECT ON V$OSSTAT TO zabbix_example;
GRANT SELECT ON V$PGASTAT TO zabbix_example;
GRANT SELECT ON V$PROCESS TO zabbix_example;
GRANT SELECT ON V$RECOVERY_FILE_DEST TO zabbix_example;
GRANT SELECT ON V$RESTORE_POINT TO zabbix_example;
GRANT SELECT ON V$SESSION TO zabbix_example;
GRANT SELECT ON V$SGASTAT TO zabbix_example;
GRANT SELECT ON V$SYSMETRIC TO zabbix_example;
GRANT SELECT ON V$SYSTEM_PARAMETER TO zabbix_example;
```

3.2 Configuration

The Zabbix agent 2 configuration file is used to configure plugins.

Plugins.Oracle.CallTimeout – The maximum amount of time, in seconds, that a request can be left unattended.

Default value: equals the global Timeout configuration parameter.

Limits: 1-30

Plugins.Oracle.ConnectTimeout – The maximum amount of time in seconds that can pass while waiting to establish a connection.

Default value: equals the global Timeout configuration parameter.

Limits: 1-30

Plugins.Oracle.CustomQueriesPath – Full pathname of a directory containing .sql files with custom queries.

Default value: (disabled by default)

Plugins.Oracle.KeepAlive – Establishes a waiting period before unused connections are closed.

Default value: 300 seconds.

Limits: 60-900

3.2.1 Connection Configuration

A connection can be configured using either keys' parameters or named sessions. Combining configuration using named sessions and key parameters at the same time is not supported. You can omit any connection parameter; in such case, a hard-coded default value will be used. Userinfo credentials that are included in URIs are forbidden and will be disregarded.

Using keys' parameters

All keys share the following parameters: [ConnString] [,User][,Password] [,Service] where ConnString may be a session name or a URI. If no session matching the specified name is identified, ConnString will be handled as a URI. Simply omit the remaining connection arguments if ConnString is used as the session name.

Using named sessions

It is possible to define particular parameters for each Oracle instance using named sessions. Only four parameters are accepted: URI, User, Password, and Service. In comparison to item keys or macros, it is a little more secure method of storing credentials.

```
Plugins.Oracle.Sessions.Oracle12.Uri=tcp://192.168.1.1:1521
Plugins.Oracle.Sessions.Oracle12.User=<USERFORORACLE12>
Plugins.Oracle.Sessions.Oracle12.Password=<PasswordForOracle12>
Plugins.Oracle.Sessions.Oracle12.Service=orcl
```

```
Plugins.Oracle.Sessions.Oracle19.Uri=tcp://192.168.1.2:1521
Plugins.Oracle.Sessions.Oracle19.User=<USERFORORACLE19>
Plugins.Oracle.Sessions.Oracle19.Password=<PasswordForOracle19>
Plugins.Oracle.Sessions.Oracle19.Service=orcl
```

3.3 Custom Queries

Using user-defined queries, the plugin's capabilities can be extended. To accomplish this, one must put all of the queries in the directory specified under `Plugins.Oracle.CustomQueriesPath` as `*.sql` files (there is no default path).

This needs to be set: `Plugins.Oracle.CustomQueriesPath=/etc/zabbix/oracle/sql`

When the queries are in place, we can execute them:

```
oracle.custom.query[<commonParams>,top_proc]
oracle.custom.query[<commonParams>,long_tx,600]
```

A query can accept as many parameters as necessary. The syntax for placeholder parameters is `":#,"` where `"#"` denotes a parameter's index number. For example, consider the following query:

```
/* payment.sql */

SELECT
    amount
FROM
    payment
WHERE
    user = :1
    AND service_id = :2
    AND date = :3
```

In the `WHERE` clause we can see the parameters `:#`, so, when the query is called these parameters will be substituted by the intended ones.

```
oracle.custom.query[<commonParams>,payment,"John Doe",1,"10/25/2020"]
```

3.4 Supported Keys

The supported keys are shown below:

oracle.diskgroups.stats[<commonParams>] – Returns ASM disk groups statistics.

oracle.diskgroups.discovery[<commonParams>] – Returns list of ASM disk groups in LLD format.

oracle.archive.info[<commonParams>] – Returns archive logs statistics.

oracle.archive.discovery[<commonParams>] – Returns list of archive logs in LLD format.

oracle.cdb.info[<commonParams>] – Returns CDBs info.

oracle.custom.query[<commonParams>,queryName[,args...]] – Returns result of a custom query.

Parameters:

queryName (required) – name of a custom query (must be equal to a name of a sql file without an extension).

args (optional) – one or more arguments to pass to a query.

oracle.datafiles.stats[<commonParams>] – Returns data files statistics.

oracle.db.discovery[<commonParams>] – Returns list of databases in LLD format.

oracle.fra.stats[<commonParams>] – Returns FRA statistics.

oracle.instance.info[<commonParams>] – Returns instance stats.

oracle.pdb.info[<commonParams>] – Returns PDBs info.

oracle.pdb.discovery[<commonParams>] – Returns list of PDBs in LLD format.

oracle.pga.stats[<commonParams>] – Returns PGA statistics.

oracle.ping[<commonParams>] – Tests if connection is alive or not.

Returns:

- "1" if a connection is alive
- "0" if a connection is broken (if there is any error presented including AUTH and configuration issues).

oracle.proc.stats[<commonParams>] – Returns processes statistics.

oracle.redolog.info[<commonParams>] – Returns log file information from the control file.

oracle.sga.stats[<commonParams>] – Returns SGA statistics.

oracle.sessions.stats[<commonParams>,[lockMaxTime]] – Returns sessions statistics.

Parameters:

lockMaxTime (optional) – maximum session lock duration in seconds to count the session as a prolonged locked.

Default – 600 seconds.

oracle.sys.metrics[<commonParams>[,duration]] – Returns a set of system metric values.

Parameters:

duration (optional) – capturing interval in seconds of system metric values. Possible values:

60 – long duration (default).

15 – short duration.

oracle.sys.params[<commonParams>] – Returns a set of system parameter values.

oracle.ts.stats[<commonParams>] – Returns tablespaces statistics.

oracle.ts.discovery[<commonParams>] – Returns list of tablespaces in LLD format.

oracle.user.info[<commonParams>[,username]] – Returns user information.

Parameters:

username (optional) – a username for which the information is needed.

Default – current user.

3.5 Oracle by Zabbix agent 2

Oracle by Zabbix Agent 2 is a template developed to monitor DBMS Oracle Database single instance via Zabbix agent 2. It was developed for Zabbix 6.2 and higher and has been tested with Oracle Database version 12c2, 18c and 19c.

To guarantee the correct operation of zabbix's templates, the following steps should be taken:

1. Ensure that the agent 2 is set up on the host and that the necessary plugin is included in the installed version. You might occasionally need to upgrade agent 2 first.
2. Link the template to a target host. To link a template, on the zabbix front-end, one must:
 - (a) Go to *Configuration* → *Hosts*
 - (b) Select the required host
 - (c) From the list of matching templates in the *Templates* field, select the intended template
 - (d) Click on *Add/Update* on the host attributes form
3. Change the obligatory macros' values. Keep in mind that user macros can overrule configuration settings.
4. Set up the monitored instance to allow data exchange with Zabbix. This step is further explained on subsection [3.1.2](#)

The template has four mandatory macros that need to be edited:

- **\$ORACLE.CONNSTRING** - Macro for the connection string. Can either be a URI or a session name defined in the format: <protocol(host:port or /path/to/socket)/>.

For URI only a TCP schema is supported.
- **\$ORACLE.SERVICE** - Macro for the Oracle Service name. The default service is ORA.

If \$ORACLE.CONNSTRING is a URI then this Macro is required.
If \$ORACLE.CONNSTRING is a named session then this macro must be empty.
- **\$ORACLE.USER** - Macro for the Oracle Username. The default username is zabbix.

If \$ORACLE.CONNSTRING is a URI then this Macro is required.
If \$ORACLE.CONNSTRING is a named session then this macro must be empty.
- **\$ORACLE.PASSWORD** - Macro for the Oracle Password. The default password is zabbix_password.

If \$ORACLE.CONNSTRING is a URI then this Macro is required.
If \$ORACLE.CONNSTRING is a named session then this macro must be empty.

To test availability run the command:

```
zabbix_get -s oracle-host -k oracle.ping["{$ORACLE.CONNSTRING}"]
→ , "{$ORACLE.USER} ", "{$ORACLE.PASSWORD} ", "{$ORACLE.SERVICE} " ]
```

3.5.1 Macros

Besides the mandatory Macros, this template also includes a vast number of Macros to avoid further configurations. Below is shown the full list. All Macros have a default value but in this project some of them might be adjusted

- **\$ORACLE.ASM.USED.PCT.MAX.HIGH**

Description: Maximum percentage of used ASM disk group for high trigger expression.
Default: 95
- **\$ORACLE.ASM.USED.PCT.MAX.WARN**

Description: Maximum percentage of used ASM disk group for warning trigger expression.
Default: 90
- **\$ORACLE.CONCURRENCY.MAX.WARN**

Description: Maximum percentage of sessions concurrency usage for trigger expression.
Default: 80
- **\$ORACLE.DB.FILE.MAX.WARN**

Description: Maximum percentage of database files for trigger expression
Default: 80

- **\$ORACLE.DBNAME.MATCHES**

Description: Used in database discovery. It can be overridden on the host or linked template level.

Default: .*

- **\$ORACLE.EXPIRE.PASSWORD.MIN.WARN**

Description: Number of days of warning before password expires (for trigger expression).

Default: 7

- **\$ORACLE.PGA.USE.MAX.WARN**

Description: Maximum percentage of PGA usage alert threshold (for trigger expression).

Default: 90

- **\$ORACLE.PROCESSES.MAX.WARN**

Description: Maximum percentage of active processes alert threshold (for trigger expression)

Default: 80

- **\$ORACLE.REDO.MIN.WARN**

Description: Minimum number of REDO logs alert threshold (for trigger expression).

Default: 3

- **\$ORACLE.SESSION.LOCK.MAX.TIME**

Description: Maximum session lock duration in seconds for count the session as a prolongedly locked query.

Default: 600

- **\$ORACLE.SESSION.LONG.LOCK.MAX.WARN**

Description: Maximum number of the prolongedly locked sessions alert threshold (for trigger expression).

Default: 3

- **\$ORACLE.SESIONS.LOCK.MAX.WARN**

Description: Maximum percentage of locked sessions alert threshold (for trigger expression).

Default: 20

- **\$ORACLE.SESIONS.MAX.WARN**

Description: Maximum percentage of active sessions alert threshold (for trigger expression).

Default: 80

- **\$ORACLE.SHARED.FREE.MIN.WARN**

Description: Minimum percentage of free shared pool alert threshold (for trigger expression).

Default: 5

- **\$ORACLE.TABLESPACE.NAME.MATCHES**

Description: Used in tablespace discovery. It can be overridden on a host or linked template level.

Default: .*

- **\$ORACLE.TABLESPACE.NAME.NOT_MATCHES**

Description: Used in tablespace discovery. It can be overridden on a host or linked template level.

Default: CHANGE_IF_NEEDED

- **\$ORACLE.TBS.USED.PCT.MAX.HIGH**

Description: Maximum percentage of used (Used bytes/Allocated bytes) tablespace high severity alert threshold (for trigger expression).

Default: 95

- **\$ORACLE.TBS.USED.PCT.MAX.WARN**

Description: Maximum percentage of used (Used bytes/Allocated bytes) tablespace warning severity alert threshold (for trigger expression).

Default: 90

- **\$ORACLE.TBS.UTIL.PCT.MAX.HIGH**

Description: Maximum percentage of utilization (Allocated bytes/Max bytes) tablespace high severity alert threshold (for trigger expression).

Default: 90

- **\$ORACLE.TBS.UTIL.PCT.MAX.WARN**

Description: Maximum percentage of utilization (Allocated bytes/Max bytes) tablespace warning severity alert threshold (for trigger expression).

Default: 80

3.5.2 Triggers

This template include various triggers to evaluate data gathered and represent the current system state. Triggers have various levels of severity, showing how important the trigger is.

SEVERITY	DEFINITION	COLOR
Not Classified	Unknown severity.	Gray
Information	For information purposes.	Light blue
Warning	Be warned.	Yellow
Average	Average problem.	Orange
High	Something important has happened.	Light red
Disaster	Disaster. Financial losses, etc.	Red

The different triggers of the template are shown below:

- **Oracle: Connection to database is unavailable**

Description: Oracle: Connection to Oracle Database is currently unavailable.

Expression:

```
last(/Oracle by Zabbix agent
→ 2/oracle.ping["${$ORACLE.CONNSTRING}","${$ORACLE.USER}",""]
→ {"${$ORACLE.PASSWORD}","${$ORACLE.SERVICE}"]=0
```

- **Oracle: Version has changed**

Description: Oracle DB version has changed. Ack to close.

Expression:

```
last(/Oracle by Zabbix agent
→ 2/oracle.version,#1)<>last(/Oracle by Zabbix agent
→ 2/oracle.version,#2) and length(last(/Oracle by Zabbix
→ agent 2/oracle.version))>0
```

- **Oracle: Failed to fetch info data**

Description: Zabbix has not received data for items for the last 5 minutes. The database might be unavailable for connecting.

Expression:

```
nodata(/Oracle by Zabbix agent 2/oracle.uptime,30m)=1
```

- **Oracle: has been restarted**

Description: Uptime is less than 10 minutes.

Expression:

```
last(/Oracle by Zabbix agent 2/oracle.uptime)<10m
```

- **Oracle: Instance name has changed**

Description: Oracle DB Instance name has changed. Ack to close.

Expression:

```

last(/Oracle by Zabbix agent
  ↪ 2/oracle.instance_name,#1)<>last(/Oracle by Zabbix
  ↪ agent 2/oracle.instance_name,#2) and
  ↪ length(last(/Oracle by Zabbix agent
  ↪ 2/oracle.instance_name))>0

```

- **Oracle: Instance hostname has changed**

Description: Oracle DB Instance hostname has changed. Ack to close.

Expression:

```

last(/Oracle by Zabbix agent
  ↪ 2/oracle.instance_hostname,#1)<>last(/Oracle by Zabbix
  ↪ agent 2/oracle.instance_hostname,#2) and
  ↪ length(last(/Oracle by Zabbix agent
  ↪ 2/oracle.instance_hostname))>0

```

- **Oracle: Shared pool free is too low**

Description: The shared pool free memory percent has been less than `{$ORACLE.SHARED.FREE.MIN.WARN}%` in the last 5 minutes.

Expression:

```

max(/Oracle by Zabbix agent 2/oracle.shared_pool_free,5m)<{
  ↪ \{$ORACLE.SHARED.FREE.MIN.WARN}

```

- **Oracle: Too many active sessions**

Description: Active sessions are using more than `{$ORACLE.SESIONS.MAX.WARN}%` of the available sessions.

Expression:

```

min(/Oracle by Zabbix agent 2/oracle.session_count,5m) *
  ↪ 100 / last(/Oracle by Zabbix agent
  ↪ 2/oracle.session_limit) > {\$ORACLE.SESIONS.MAX.WARN}

```

- **Oracle: Too many locked sessions**

Description: Number of locked sessions is over `{$ORACLE.SESIONS.LOCK.MAX.WARN}%` of the running sessions.

Expression:

```

min(/Oracle by Zabbix agent 2/oracle.session_lock_rate,5m)
  ↪ > {\$ORACLE.SESIONS.LOCK.MAX.WARN}

```

- **Oracle: Too many sessions locked**

Description: Number of sessions locked over `$ORACLE.SESSION.LOCK.MAX.TIME` seconds is too high. Long-term locks can negatively affect database performance, therefore, if they are detected, you should first find the most difficult queries from the database point of view and analyze possible resource leaks.

Expression:

```
min(/Oracle by Zabbix agent
  → 2/oracle.session_long_time_locked,5m) >
  → {$ORACLE.SESSION.LONG.LOCK.MAX.WARN}
```

- **Oracle: Too high database concurrency**

Description: Concurrency rate is over `{$ORACLE.CONCURRENCY.MAX.WARN}%`. A high contention value does not indicate the root cause of the problem, but is a signal to search for it. In the case of high competition, an analysis of resource consumption should be carried out, the most "heavy" queries made in the database, possibly - session tracing. All this will help determine the root cause and possible optimization points both in the database configuration and in the logic of building queries of the application itself.

Expression:

```
min(/Oracle by Zabbix agent
  → 2/oracle.session_concurrency_rate,5m) >
  → {$ORACLE.CONCURRENCY.MAX.WARN}
```

- **Oracle: Total PGA in use is too high**

Description: Total PGA in use is more than `{$ORACLE.PGA.USE.MAX.WARN}%` of `PGA_AGGREGATE_TARGET`.

Expression:

```
min(/Oracle by Zabbix agent 2/oracle.total_pga_used,5m) *
  → 100 / last(/Oracle by Zabbix agent 2/oracle.pga_target)
  → > {$ORACLE.PGA.USE.MAX.WARN}
```

- **Oracle: Zabbix account will expire soon**

Description: Password for zabbix user in the database will expire soon.

Expression:

```
last(/Oracle by Zabbix agent
  → 2/oracle.user.info["{$ORACLE.CONNSTRING}", "{$ORACLE.USER}_
  → R", "{$ORACLE.PASSWORD}", "{$ORACLE.SERVICE}"]) <
  → {$ORACLE.EXPIRE.PASSWORD.MIN.WARN}
```

- **Oracle: Number of REDO logs available for switching is too low**

Description: Number of available for log switching inactive/unused REDOs is low (Database down risk)

Expression:

```
max(/Oracle by Zabbix agent
  ↪ 2/oracle.redolog.info["${ORACLE.CONNSTRING}", "${ORACLE.US
  ↪ USER}", "${ORACLE.PASSWORD}", "${ORACLE.SERVICE}"], 5m) <
  ↪ ${ORACLE.REDO.MIN.WARN}
```

- **Oracle: Too many active processes**

Description: Active processes are using more than \${ORACLE.PROCESSES.MAX.WARN}% of the available number of processes.

Expression:

```
min(/Oracle by Zabbix agent
  ↪ 2/oracle.proc.stats["${ORACLE.CONNSTRING}", "${ORACLE.US
  ↪ ER}", "${ORACLE.PASSWORD}", "${ORACLE.SERVICE}"], 5m) *
  ↪ 100 / last(/Oracle by Zabbix agent
  ↪ 2/oracle.processes_limit) >
  ↪ ${ORACLE.PROCESSES.MAX.WARN}
```

- **Oracle: Too many database files**

Description: Number of datafiles is higher than \${ORACLE.DB.FILE.MAX.WARN}% of the available datafile files limit.

Expression:

```
min(/Oracle by Zabbix agent 2/oracle.datafiles.stats["{
  ↪ $ORACLE.CONNSTRING}", "${ORACLE.USER}", "${ORACLE.PAS
  ↪ SWORD}", "${ORACLE.SERVICE}"], 5m) * 100 /
  ↪ last(/Oracle by Zabbix agent
  ↪ 2/oracle.db_files_limit) >
  ↪ ${ORACLE.DB.FILE.MAX.WARN}
```

- **Oracle Database '#DBNAME': Open status in mount mode**

Description: The Oracle DB has a MOUNTED state.

Expression:

```
last(/Oracle by Zabbix agent
  ↪ 2/oracle.db_open_mode["${#DBNAME}"])=1
```

- **Oracle Database '#DBNAME': Open status has changed**

Description: Oracle DB open status has changed. Ack to close.

Expression:

```
last(/Oracle by Zabbix agent 2/oracle.db_open_mode["#{DB_
→ BNAME}"],#1)<>last(/Oracle by Zabbix agent
→ 2/oracle.db_open_mode["#{DBNAME}"],#2)
```

- **Oracle Database '#DBNAME': Role has changed**

Description: Oracle DB role has changed. Ack to close.

Expression:

```
last(/Oracle by Zabbix agent
→ 2/oracle.db_role["#{DBNAME}"],#1)<>last(/Oracle by
→ Zabbix agent 2/oracle.db_role["#{DBNAME}"],#2)
```

- **Oracle Database '#DBNAME': Force logging is deactivated for DB with active ArchiveLog**

Description: Force Logging mode - it is very important metric for Databases in 'ARCHIVELOG'. This feature allows to forcibly write all transactions to the REDO.

Expression:

```
last(/Oracle by Zabbix agent
→ 2/oracle.db_force_logging["#{DBNAME}"]) = 0 and
→ last(/Oracle by Zabbix agent
→ 2/oracle.db_log_mode["#{DBNAME}"]) = 1
```

- **Oracle Database '#DBNAME': Open status in mount mode**

Description: The Oracle DB has a MOUNTED state.

Expression:

```
last(/Oracle by Zabbix agent
→ 2/oracle.pdb_open_mode["#{DBNAME}"]=1
```

- **Oracle Database '#DBNAME': Open status has changed**

Description: Oracle DB open status has changed. Ack to close.

Expression:

```
last(/Oracle by Zabbix agent 2/oracle.pdb_open_mode["#{_
→ DBNAME}"],#1)<>last(/Oracle by Zabbix agent
→ 2/oracle.pdb_open_mode["#{DBNAME}"],#2)
```

- **Oracle TBS '#TABLESPACE': Tablespace usage is too high**

Expression:

```
min(/Oracle by Zabbix agent
→ 2/oracle.tbs_used_file_pct["#{TABLESPACE}"],5m)>{$_O_
→ RACLE.TBS.USED.PCT.MAX.WARN}
```


- **Oracle TBS '#TABLESPACE': Tablespace usage is too high**

Expression:

```
min(/Oracle by Zabbix agent
  ↪ 2/oracle.tbs_used_file_pct["#{TABLESPACE}"],5m)>{$ORACLE.TBS.USED.PCT.MAX.HIGH}
```

- **Oracle TBS '#TABLESPACE': Tablespace utilization is too high**

Expression:

```
min(/Oracle by Zabbix agent 2/oracle.tbs_used_pct["#{TABLESPACE}"],5m)>{$ORACLE.TBS.USED.PCT.MAX.WARN}
```

- **Oracle TBS '#TABLESPACE': Tablespace utilization is too high**

Expression:

```
min(/Oracle by Zabbix agent 2/oracle.tbs_used_pct["#{TABLESPACE}"],5m)>{$ORACLE.TBS.UTIL.PCT.MAX.HIGH}
```

- **Oracle TBS '#TABLESPACE': Tablespace is OFFLINE**

Description: The tablespace is in the offline state.

Expression:

```
last(/Oracle by Zabbix agent
  ↪ 2/oracle.tbs_status["#{TABLESPACE}"])=2
```

- **Oracle TBS '#TABLESPACE': Tablespace status has changed**

Description: Oracle tablespace status has changed. Ack to close.

Expression:

```
last(/Oracle by Zabbix agent 2/oracle.tbs_status["#{TABLESPACE}"],#1)<>last(/Oracle by Zabbix agent
  ↪ 2/oracle.tbs_status["#{TABLESPACE}"],#2)
```

- **Archivelog '#DEST_NAME': Log Archive is not valid**

Description: ARL destination not in 3 - Valid or 2 - Deferred.

Expression:

```
last(/Oracle by Zabbix agent
  ↪ 2/oracle.archivelog_log_status["#{DEST_NAME}"])<2
```

- **ASM '#DG_NAME': Disk group usage is too high**

Description: Usage percent of ASM disk group is over {\$ORACLE.ASM.USED.PCT.MAX.WARN}

Expression:

```
lmin(/Oracle by Zabbix agent 2/oracle.asm_used_pct["{#D}
→ G_NAME}"], 5m) > {$ORACLE.ASM.USED.PCT.MAX.WARN}
```

- **ASM '#DG_NAME': Disk group usage is too high**

Description: Usage percent of ASM disk group is over {\$ORACLE.ASM.USED.PCT.MAX.WARN}

Expression:

```
min(/Oracle by Zabbix agent 2/oracle.asm_used_pct["{#DG}
→ _NAME}"], 5m) > {$ORACLE.ASM.USED.PCT.MAX.HIGH}
```

3.5.3 Items

All the items collected by this template are now shown on this subsection. These items are very important, since they provide diverse information about the current state and status of various aspects of the database being monitored.

Some items may then be connected to triggers, if the value received is not expected and should be different.

3.5.3.1 Database Application

Table 3.1 list all items collected related to the database application. It shows more generic information like the database instance information and user related metrics.

Even though they are more generic they are still of high value to the DBA and should have as much attention as the other more specific items.

3.5.3.2 Memory

Oracle memory is divided in two main areas, called System Global memory Area (SGA) and Program Global Area (PGA). The SGA's memory usage is predetermined at launch and is shown in the alert log. AWR and STATSPACK reports will display Oracle memory consumption with a starting memory usage and ending memory usage if you are using automated memory management (not advised for large databases).

Items related to the SGA are shown on table 3.2

Now, when it comes to PGA, excessive PGA usage can lead to memory swapping issues and have impact on the performance. So, this items provide important information for the administrators. The PGA_AGGREGATE_TARGET is the parameter for control PGA memory allocation. Table 3.3 shows all PGA items collected

3.5.3.3 FRA

Fast Recovery Area (FRA) provides a centralized storage location for backup and recovery files. Oracle creates archived logs and flashback logs in the fast recovery area. It is important to monitor space usage in the fast recovery area to ensure that it is large enough to contain backups and other recovery-related files.

3.5.3.4 #TABLESPACE

The logical database structures for an Oracle Database are called tablespaces. Each tablespace designates the data files that contain the actual data. Indexes, tables, and other database objects can all be found in an Oracle tablespace. Tablespace monitoring is crucial, thus the database administrator must do it.

The Oracle DBA must keep track of whether the tablespace is full and whether any capacity-boosting measures are necessary. You can add a datafile to an existing tablespace, create a new tablespace, or even increase the size of the datafiles that are already present in a tablespace to improve capacity.

Table 3.5 shows the items collected regarding Tablespaces

3.5.3.5 #DBNAME

The items collected in this section are related to the database itself. they show some settings of the database like the open status and the current role.

All the items are more detailed in the table 3.6.

3.5.3.6 Sessions

Every time a user connects from a particular process to the Oracle database, the connection is considered a session. We can monitor a server's activity by counting the number of Oracle sessions that are running on it. This provides information on the load on an Oracle Database. Additionally, it aids in identifying which users consume the most system resources.

For the purpose of guaranteeing the efficient operation of production databases, it is crucial for the database administrator to monitor the sessions.

Table 3.7 shows all items collected that are related to sessions.

3.5.3.7 Cache

By caching frequently used data on a middle-tier machine, Oracle Database Cache increases the scalability and speed of applications that access Oracle databases. Applications can handle several times as many queries as they could before using Oracle Database Cache.

Monitoring the cache is very important since it provides information about which queries are hits and which are misses, as well as information about hits and misses per second, processor time, and memory usage.

A hit is a query that was satisfied in the middle-tier cache. A miss is a query that could not be satisfied in the cache and that was routed to the origin database.

Table 3.8 shows all items collected regarding the Oracle cache.

3.5.3.8 CPU

Monitoring the CPU usage is very important towards a good performance of the database. The Database CPU Time Ratio is calculated by dividing the total CPU used by the database by the amount of total database time. Total database time is the time spent by the database on user-level calls (without instance background process activity).

Values above 90% are generally good and indicate few waits or bottlenecks. A lower value for Database CPU Time Ratio only means that the processing has changed from CPU to disk, network or other system time.

Table 3.9 shows all items collected that are related to the CPU

3.5.3.9 Disk

The metric Physical Reads shows how many data blocks were read from the disk each second during a sampling period. When a user issues a SQL query, Oracle initially looks to obtain the data from the database buffer cache (memory), and if that does not succeed, it scans the disk. It takes substantially longer to read data blocks from disk than it does to read them from memory. Oracle should constantly want to use as much memory as possible.

The metric Physical Write represents the number of disk writes per second during the sample period. This statistic represents the rate of database blocks written from the SGA buffer cached to disk by the DBWR (Database Writer) background process, and from the PGA by processes performing direct writes.

These are of very high importance in terms of efficiency.

Table 3.10 shows all items collected regarding the Disk.

3.5.3.10 System

The DBA should use the information provided by the processes items collected and evaluate whether the database could use some efficiency updates or not.

The template collects the number of processes and the processes limit as shown in table 3.11.

3.5.3.11 Datafiles

Datafiles are the actual physical structure where data is physically kept by the Oracle Database, whereas Oracle tablespaces are the logical structures. The physical storage location of the data, indicated by a tablespace, may be one or more datafiles. In order to ensure that databases run well, it is crucial for the database administrator to monitor datafiles in addition to tablespaces.

Table 3.12 shows all items collected related to the datafiles.

3.5.3.12 Raw

Besides the list of items previously detailed above, this template also collects some raw items like statistics items and CDB information. Table 3.13 details all these items.

Name	Description
Archiver state	Automatic archiving status.
Database wait time ratio	Wait time: the time that the server process spends waiting for available shared resources (to be released by other server processes) such as latches, locks, data buffers, and so on
Disk sort per second	The number of sorts going to disk per second
Enqueue timeouts per second	Enqueue timeouts per second.
Uptime	Oracle instance uptime in seconds.
Instance Status	Status of the instance.
Instance Name	Name of the instance.
Instance Hostname	Name of the host machine.
Instance Role	Indicates whether the instance is an active instance or an inactive secondary instance.
Get instance state	The item gets state of the current instance.
Logons per second	The number of logon attempts.
Long table scans per second	The number of long table scans per second. A table is considered 'long' if the table is not cached and if its high-water mark is greater than 5 blocks.
Memory sorts ratio	The percentage of sorts (from ORDER BY clauses or index building) that are done to disk vs in-memory.
Ping	Test the connection to Oracle Database state
Redo logs available to switch	Number of available for log switching inactive/unused REDO logs.
Rows per sort	The average number of rows per sort for all types of sorts performed.
SQL service response time	SQL service response time in seconds.
Temp space used	Temporary space used.
Total sorts per user call	Total sorts per user call.
User rollbacks per second	The number of times that users manually issue the ROLLBACK statement or an error occurred during a user's transactions.
User's expire password	The number of days before zabbix account password expired.
Oracle: Version	Oracle Server version.

Table 3.1: Information collected about the database application

Name	Description
Java pool	Memory is allocated from the java pool.
Large pool	Memory is allocated from the large pool.
Shared pool	Memory is allocated from the shared pool.
Shared pool free %	Shared pool free memory percent. Free/Total
Log buffer	The number of bytes allocated for the redo log buffer.
Fixed	The fixed SGA is an internal housekeeping area.

Table 3.2: Information collected about SGA

Name	Description
Total inuse	Indicates how much PGA memory is currently consumed by work areas. This number can be used to determine how much memory is consumed by other consumers of the PGA memory (for example, PL/SQL or Java).
Aggregate target parameter	Current value of the PGA_AGGREGATE_TARGET initialization parameter. If this parameter is not set, then its value is 0 and automatic management of PGA memory is disabled.
Total allocated	Current amount of PGA memory allocated by the instance. The Oracle Database attempts to keep this number below the value of the PGA_AGGREGATE_TARGET initialization parameter. However, it is possible for the PGA allocated to exceed that value by a small percentage and for a short period of time when the work area workload is increasing very rapidly or when PGA_AGGREGATE_TARGET is set to a small value.
Total freeable	Number of bytes of PGA memory in all processes that could be freed back to the operating system.
Global memory bound	Maximum size of a work area executed in automatic mode.

Table 3.3: Information collected about PGA

Name	Description
Space limit	Maximum amount of disk space (in bytes) that the database can use for the fast recovery area.
Used space	Amount of disk space (in bytes) used by fast recovery area files created in current and all previous fast recovery areas.
Space reclaimable	Total amount of disk space (in bytes) that can be created by deleting obsolete, redundant, and other low priority files from the fast recovery area.
Number of files	Number of files in the fast recovery area
Usable space in %	Percentage of usable space in the FRA
Number of restore points	Number of restore points

Table 3.4: Information collected about FRA

Name	Description
Tablespace allocated, bytes	Currently allocated bytes for tablespace (sum of the current size of datafiles).
Tablespace MAX size, bytes	Maximum size of tablespace.
Tablespace used, bytes	Currently used bytes for tablespace (current size of datafiles-free space).
Tablespace free, bytes	Used bytes/Allocated bytes*100
Tablespace usage, percent	Indicates whether the instance is an active instance or an inactive secondary instance.
Tablespace allocated, percent	Allocated bytes/Max bytes*100
Open status	Shows tablespaces status. 1 - 'ONLINE' 2 - 'OFFLINE' 3- 'READ ONLY'
Get Tablespaces stats	Shows Tablespaces statistics

Table 3.5: Information collected about the #TABLESPACES

Name	Description
Open status	1 - 'MOUNTED', 2 - 'READ WRITE', 3 - 'READ ONLY', 4 - 'READ ONLY WITH APPLY' (A physical standby database is open in real-time query mode)
Role	Current role of the database, 1 - 'SNAPSHOT STANDBY', 2 - 'LOGICAL STANDBY', 3 - 'PHYSICAL STANDBY', 4 - 'PRIMARY', 5 - 'FAR SYNC'
Log mode	Archive log mode, 0 - 'NOARCHIVELOG', 1 - 'ARCHIVELOG', 2 - 'MANUAL'
Force logging	Indicates whether the database is under force logging mode (YES) or not (NO)

Table 3.6: Information collected about the #DBNAME

Name	Description
Average active sessions	The average active sessions at a point in time. It is the number of sessions that are either working or waiting.
Active serial sessions	The number of active serial sessions.
Active parallel sessions	The number of active parallel sessions.
Sessions limit	User and system sessions.
Session count	Session count
Active user sessions	The number of active user sessions.
Active background sessions	The number of active background sessions.
Library cache hit ratio	Ratio of library cache hits. Hits/Pins.
Inactive user sessions	The number of inactive user sessions.
Sessions lock rate	The percentage of locked sessions. Locks are mechanisms that prevent destructive interaction between transactions accessing the same resource—either user objects such as tables and rows or system objects not visible to users, such as shared data structures in memory and data dictionary rows.
Sessions locked for a long time	Count of the prolongedly locked sessions. (You can change maximum session lock duration in seconds for query by <code>\$ORACLE.SESSION.LOCK.MAX.TIME</code> macro. Default 600 sec)
Sessions concurrency	The percentage of concurrency. Concurrency is a DB behavior when different transactions request to change the same resource - in case of modifying data transactions sequentially block temporarily the right to change data, the rest of the transactions are waiting for access. In the case when access for resource is locked for a long time, then the concurrency grows (like the transaction queue) and this often has an extremely negative impact on performance. A high contention value does not indicate the root cause of the problem, but is a signal to search for it.
Get sessions stats	Get sessions statistics. <code>\$ORACLE.SESSION.LOCK.MAX.TIME</code> – maximum seconds in the current wait condition for counting long time locked sessions. Default: 600 seconds.

Table 3.7: Information collected about the sessions

Name	Description
Buffer cache hit ratio	Ratio of buffer cache hits. (LogRead - PhyRead)/LogRead
Cursor cache hit ratio	Ratio of cursor cache hits. CursorCacheHit/SoftParse
GC CR block received per second	GC CR block received per second.
Global cache blocks corrupted	The number of blocks that encountered a corruption or checksum failure during interconnect.
Global cache blocks lost	The number of global cache blocks lost
SGA, buffer cache	The size of the cache of standard blocks.

Table 3.8: Cache-related Items

Name	Description
Database CPU time ratio	Calculated by dividing the total CPU used by the database by the Oracle time model statistic DB time.

Table 3.9: Information related to the CPU

Name	Description
Physical reads per second	Reads per second.
Physical writes per second	Writes per second.
Physical reads bytes per second	Read bytes per second.
Physical writes bytes per second	Write bytes per second.

Table 3.10: Information related to the Disk

Name	Description
Number of processes	Number of processes
Processes limit	Max user processes.

Table 3.11: Information related to the System

Name	Description
Datafiles count	Current number of datafile.
Datafiles limit	Max allowable number of datafile.

Table 3.12: Information related to the Datafiles

Name	Description
Get system metrics	The item gets system metric values.
Get system parameters	Get a set of system parameter values.
Get CDB and No-CDB info	Get information about CDB and No-CDB databases on instance.
Get PDB info	Get information about PDB databases on instance.
Get archive log info	Get information about the archive log
Get ASM stats	Get ASM disk groups stats.
Get FRA stats	Get FRA statistics.
Get PGA stats	Get PGA statistics.
Get SGA stats	Get SGA statistics.
Get tablespaces stats	Get tablespaces stats.
Get sessions stats	Get sessions statistics. \$ORACLE.SESSION.LOCK.MAX.TIME – maximum seconds in the current wait condition for counting long time locked sessions. Default: 600 seconds.

Table 3.13: Raw Items collected

Chapter 4

Obtained Results

This chapter will cover the results obtained and will show how Zabbix works with the Oracle template.

After everything was installed and the template linked properly to the Zabbix server, Zabbix immediately started monitoring the Oracle database. Some values were changing every few seconds while others maintained their value more time. It all depends on the refresh rate of these items. As shown on the previous chapter, some items refresh their value every 30 seconds, while others only refresh after a few minutes. It is important to know that some items have more importance than others, so monitoring those more often is a valuable task for the database administrator (DBA). Figures 4.1, 4.2, 4.3, 4.4 and 4.5 show all items being correctly collected. We can see when the item was last checked and the change between the current and the last check. This is also important to see if significant changes occurred in the database.

Host	Name	Last check	Last value	Change	Tags
francisco	Oracle: Active background sessions	33s	49		component: sessions
francisco	Oracle: Active parallel sessions	6s	0		component: sessions
francisco	Oracle: Active serial sessions	6s	1		component: sessions
francisco	Oracle: Active user sessions	33s	1		component: sessions
francisco	Oracle: Archiver state	39s	Stopped (1)		component: application
francisco	Oracle: Average active sessions	6s	0.001	-0.001	component: sessions
francisco	Oracle: Buffer cache hit ratio	6s	100 %	+0.271 %	component: cache
francisco	Oracle: Cursor cache hit ratio	6s	83.333 %	+7.333 %	component: cache
francisco	Oracle: Database CPU time ratio	6s	66.198 %	-22.452 %	component: cpu
francisco	Oracle: Database wait time ratio	6s	33.802 %	+22.452 %	component: application
francisco	Oracle: Datafiles count	42s	4		component: datafiles
francisco	Oracle: Datafiles limit	30s	200		component: datafiles
francisco	Oracle: Disk sort per second	6s	0 Rps		component: application
francisco	Oracle: Enqueue timeouts per second	6s	0 Rps		component: application
francisco	Oracle: FRA, Number of files	40s	0		component: fra
francisco	Oracle: FRA, Number of restore points	40s	0		component: fra
francisco	Oracle: FRA, Space limit	40s	0 B		component: fra
francisco	Oracle: FRA, Space reclaimable	40s	0 B		component: fra
francisco	Oracle: FRA, Usable space in %	40s	0 %		component: fra
francisco	Oracle: FRA, Used space	40s	0 B		component: fra
francisco	Oracle: GC CR block received per second	6s	0 Rps		component: cache

Figure 4.1: Items collected 1/5

francisco	Oracle: Global cache blocks corrupted	36s	0 Rps		component: cache
francisco	Oracle: Global cache blocks lost	36s	0 Rps		component: cache
francisco	Oracle: Inactive user sessions	3s	2		component: sessions
francisco	Oracle: Instance hostname	9s	8f0ed2df97ef		component: application
francisco	Oracle: Instance name	9s	ORCL18		component: application
francisco	Oracle: Instance role	9s	Primary (1)		component: application
francisco	Oracle: Instance status	9s	Open (3)		component: application
francisco	Oracle: Library cache hit ratio	36s	100 %		component: cache
francisco	Oracle: Logons per second	36s	0.017 Rps	-0.034 Rps	component: application
francisco	Oracle: Long table scans per second	36s	0 Rps		component: application
francisco	Oracle: Memory sorts ratio	36s	100 %		component: application
francisco	Oracle: Number of processes	5s	57		component: system
francisco	Oracle: PGA, Aggregate target parameter	7s	512 MB		component: memory component: pga
francisco	Oracle: PGA, Global memory bound	7s	100 MB		component: memory component: pga
francisco	Oracle: PGA, Total allocated	7s	214.65 MB		component: memory component: pga
francisco	Oracle: PGA, Total freeable	7s	24.63 MB		component: memory component: pga
francisco	Oracle: PGA, Total inuse	7s	173.61 MB	-1 KB	component: memory component: pga

Figure 4.2: Items collected 2/5

francisco	Oracle: Session count	49s	52		component: sessions
francisco	Oracle: Sessions concurrency	49s	0 %		component: sessions
francisco	Oracle: Sessions limit	46s	472		component: sessions
francisco	Oracle: Sessions locked over { <code>\$ORACLE_SESSION_LOCK_MAX_TIME</code> }s	49s	0		component: sessions
francisco	Oracle: Sessions lock rate	49s	0 %		component: sessions
francisco	Oracle: SGA, buffer cache	48s	1.03 GB		component: cache component: sga
francisco	Oracle: SGA, fixed	48s	8.48 MB		component: memory component: sga
francisco	Oracle: SGA, java pool	48s	16 MB		component: memory component: sga
francisco	Oracle: SGA, large pool	48s	32 MB		component: memory component: sga
francisco	Oracle: SGA, log buffer	48s	7.52 MB		component: memory component: sga
francisco	Oracle: SGA, shared pool	48s	262.42 MB		component: memory component: sga
francisco	Oracle: Shared pool free %	23s	13.211 %		component: memory
francisco	Oracle: SQL service response time	23s	0.08ms	+0.02ms	component: application
francisco	Oracle: Temp space used	23s	0 B		component: application
francisco	Oracle: Total sorts per user call	23s	2.682		component: application
francisco	Oracle: Uptime	55s	4h 47m 53s	+1m	component: application
francisco	Oracle: User's expire password	44s	180 days		component: application
francisco	Oracle: User rollbacks per second	23s	0 Rps		component: application
francisco	Oracle: Version	2h 5m 55s	18.0.0.0.0-EE		component: application

Figure 4.3: Items collected 3/5

francisco	Oracle TBS 'SYSAUX': Open status	46s	Online (1)		component: tablespa... contents: PERMANE... tablespace: SYSAUX
francisco	Oracle TBS 'SYSAUX': Tablespace allocated, bytes	46s	520 MB		component: tablespa... contents: PERMANE... tablespace: SYSAUX
francisco	Oracle TBS 'SYSAUX': Tablespace allocated, percent	46s	1.59 %		component: tablespa... contents: PERMANE... tablespace: SYSAUX
francisco	Oracle TBS 'SYSAUX': Tablespace free, bytes	46s	25.5 MB		component: tablespa... contents: PERMANE... tablespace: SYSAUX
francisco	Oracle TBS 'SYSAUX': Tablespace MAX size, bytes	46s	32 GB		component: tablespa... contents: PERMANE... tablespace: SYSAUX
francisco	Oracle TBS 'SYSAUX': Tablespace usage, percent	46s	95.1 %		component: tablespa... contents: PERMANE... tablespace: SYSAUX
francisco	Oracle TBS 'SYSAUX': Tablespace used, bytes	46s	494.5 MB		component: tablespa... contents: PERMANE... tablespace: SYSAUX
francisco	Oracle TBS 'SYSTEM': Open status	46s	Online (1)		component: tablespa... contents: PERMANE... tablespace: SYSTEM
francisco	Oracle TBS 'SYSTEM': Tablespace allocated, bytes	46s	840 MB		component: tablespa... contents: PERMANE... tablespace: SYSTEM
francisco	Oracle TBS 'SYSTEM': Tablespace allocated, percent	46s	2.56 %		component: tablespa... contents: PERMANE... tablespace: SYSTEM
francisco	Oracle TBS 'SYSTEM': Tablespace free, bytes	46s	6.38 MB		component: tablespa... contents: PERMANE... tablespace: SYSTEM
francisco	Oracle TBS 'SYSTEM': Tablespace MAX size, bytes	46s	32 GB		component: tablespa... contents: PERMANE... tablespace: SYSTEM
francisco	Oracle TBS 'SYSTEM': Tablespace usage, percent	46s	99.24 %		component: tablespa... contents: PERMANE... tablespace: SYSTEM
francisco	Oracle TBS 'SYSTEM': Tablespace used, bytes	46s	833.63 MB		component: tablespa... contents: PERMANE... tablespace: SYSTEM

Figure 4.4: Items collected 4/5

francisco	Oracle Database 'ORCL18': Force logging	13m	No (0)	component: database database: ORCL18 type: CDB
francisco	Oracle Database 'ORCL18': Log mode	13m	NOARCHIVELOG (0)	component: database database: ORCL18 type: CDB
francisco	Oracle Database 'ORCL18': Open status	13m	Read-Write (3)	component: database database: ORCL18 type: CDB
francisco	Oracle Database 'ORCL18': Role	13m	Primary (4)	component: database database: ORCL18 type: CDB

Figure 4.5: Items collected 5/5

Some collected items can also be viewed as a chart. These charts are very helpful to let DBA know how the item collected evolved over a time span. Figures 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 and 4.13 show examples of some item charts collected by the current Zabbix configuration.

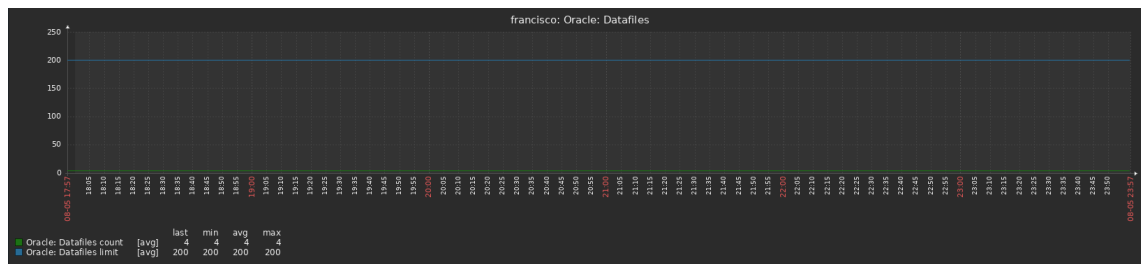


Figure 4.6: Graph about the Datafiles

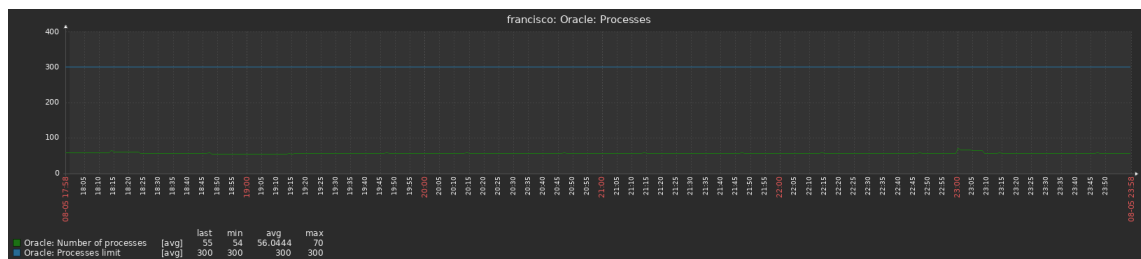


Figure 4.7: Graph about the Processes

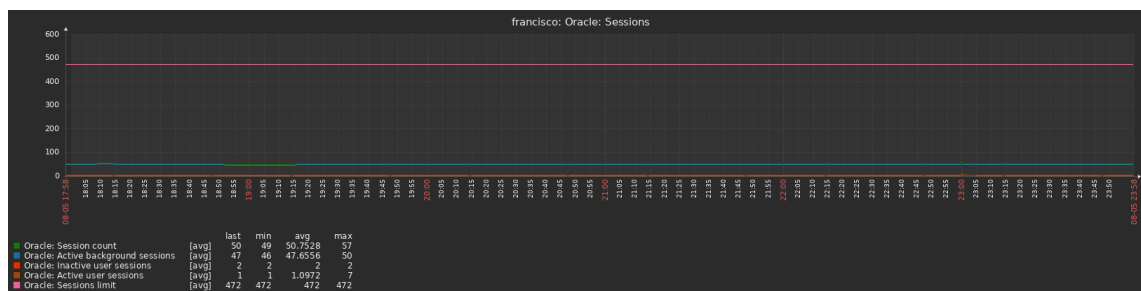


Figure 4.8: Graph about the Sessions

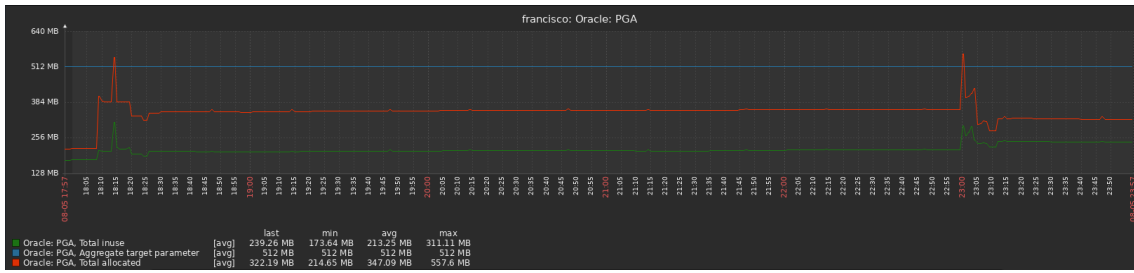


Figure 4.9: Graph about the PGA

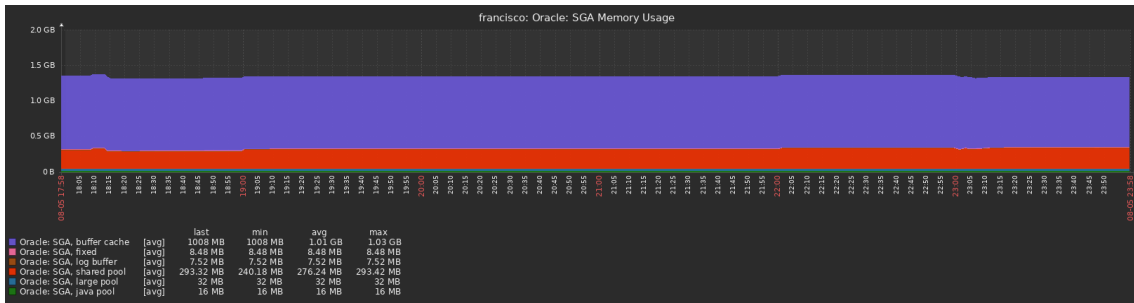


Figure 4.10: Graph about the SGA

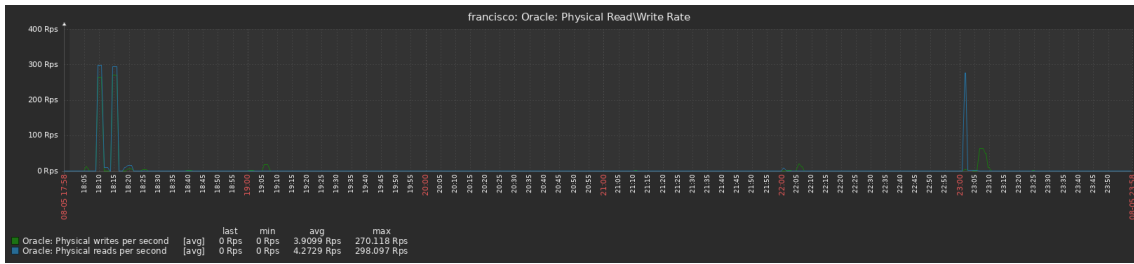


Figure 4.11: Graph about the Physical Read Write Rate

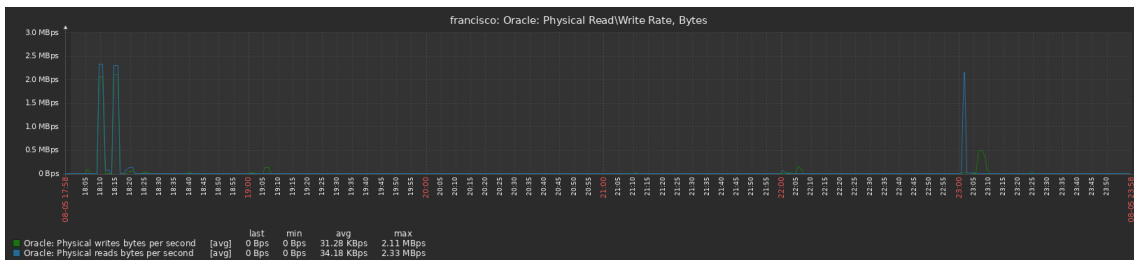


Figure 4.12: Graph about the Physical Read Write Rate in bytes

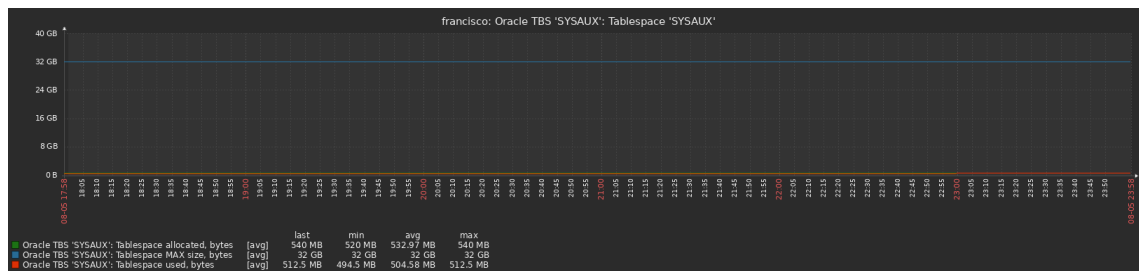


Figure 4.13: Graph about the Tablespace 'SYSUAUX'

To show how the trigger function works, some triggers were forced to appear, just to exemplify on this dissertation. On picture 4.14 we can see the Zabbix dashboard showing 4 trigger messages. One of them is not related to the database but to the machine hosting the DBMS. It is warning the manager that the space left on the disk is below 80% (the trigger value). I left it here to demonstrate that Zabbix is not only monitoring the Oracle database, but it is also monitoring the machine by default.

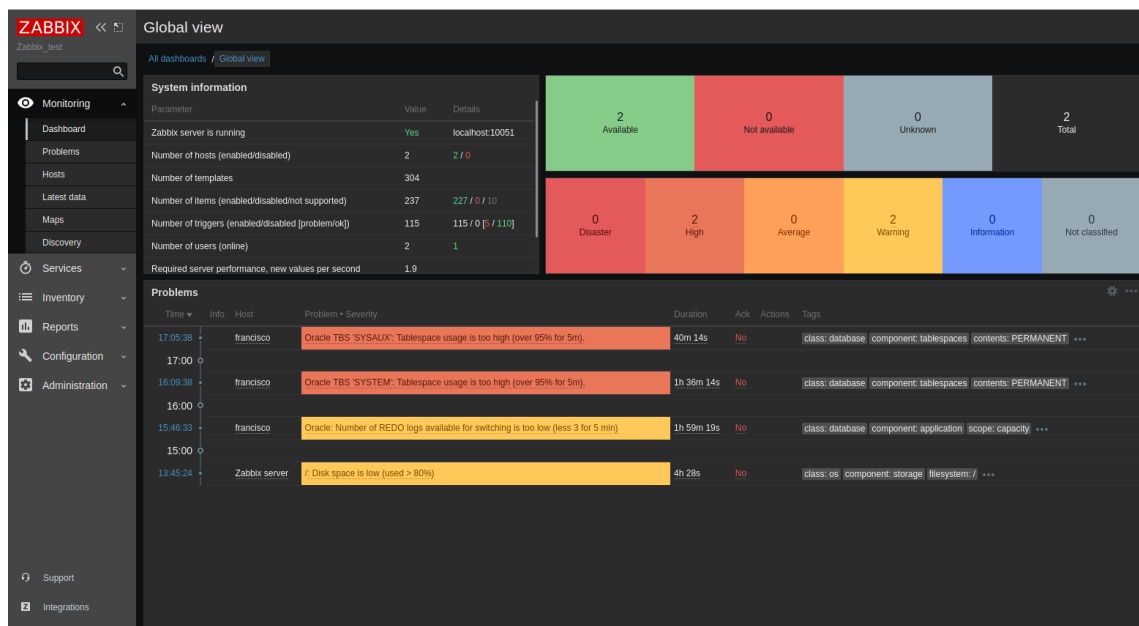


Figure 4.14: Zabbix dashboard, including alerts raised by triggers

The other 3 triggers are all related to the database. The warning was not intentionally forced and it is telling how the number of REDO logs is too low. As mentioned earlier, warnings are not as important as other triggers, but they can lead to some other problems so the DBA should have a look at this warning. The two remaining triggers were the ones forced and they are of high importance. They both are the same trigger but for different tablespaces. They are alerting that the SYSUAUX and the SYSTEM tablespaces are reaching their full space capacity (above 95%). As you can imagine, letting system tablespaces running out of space, can cause serious problems on the database.

Chapter 5

Conclusion

This project was a challenge to me. There is few information about the topic and the template created by Zabbix doesn't have very useful information on how to install, how to use and how to modify the template. The connection between the source code and the template itself is also not explained in much detail which made complicated the task of modifying the plugin to allow RAC awareness.

Despite its difficulty, this project was very rewarding. The importance of IT monitoring was something I had never thought about and made me conclude that without these monitoring tools, the efficiency of many companies would much lower and the number of problems would be much higher.

I now know that I learned to use a new tool for my work life that is a valuable asset in today's world.

5.1 Future Work

Initially this project had additional work intended but unfortunately, due to some unpredicted causes I wasn't able to complete it.

The ability to run the template on a oracle RAC system is not available. I was supposed to fix this problems by using the GV\$ views mentioned earlier, since these views retrieve the normal information but also indicate which instance they are related to.

The other problem I was supposed to fix, is related to the free space monitoring. The template was supposed to be edited to monitor the free space according to a query created previously:

```
SELECT
    to_char(sysdate, 'YYYY-MM-DD HH24:MI:SS') timest
    , tablespace_name
    , asm_diskgroup
    , round(ocup/1024/1024,0) ocup_mb
    , trunc(nvl(free/1024/1024,0)) free_mb
    , percent
```

```

, nvl(trunc(autoext/1024/1024),0) max_autoext_mb
, virtualpercent
, (SELECT FREE_MB FROM V$ASM_DISKGROUP WHERE
↪ NAME=asm_diskgroup) asm_freemb

FROM (
  WITH freespace AS (SELECT tablespace_name,sum(bytes) bytes
↪ FROM dba_free_space GROUP BY tablespace_name)
, totspace AS (SELECT tablespace_name,
↪ substr(file_name,2,instr(file_name,'/'))-2)
↪ asm_diskgroup,sum(bytes) bytes FROM dba_data_files
↪ GROUP BY tablespace_name,
↪ substr(file_name,2,instr(file_name,'/'))-2))
, autoext AS (SELECT TABLESPACE_NAME, sum(MAXBYTES-bytes)
↪ bytes FROM dba_data_files WHERE AUTOEXTENSIBLE='YES'
↪ AND status='AVAILABLE' GROUP BY TABLESPACE_NAME)

SELECT
  tbs.tablespace_name,asm_diskgroup
  , totspace.bytes ocup
  , freespace.bytes free
  , round((totspace.bytes-nvl(freespace.bytes,0))*100,
↪ /totspace.bytes,2)
  ↪ percent
  , autoext.bytes autoext
  , round((totspace.bytes-nvl(freespace.bytes,0))*100,
↪ /(totspace.bytes+nvl(autoext.bytes,0)),2)
  ↪ virtualpercent

FROM dba_tablespaces tbs

LEFT JOIN freespace ON
↪ freespace.tablespace_name=tbs.tablespace_name
LEFT JOIN totspace ON
↪ totspace.tablespace_name=tbs.tablespace_name
LEFT JOIN autoext ON
↪ autoext.tablespace_name=tbs.tablespace_name

)

```

```
ORDER BY 2,3;
```

This query allows more exact and efficient free space monitoring.

The two topics mentioned in this subsection are the logical next steps in the work presented in this dissertation.

References

- [1] Josune Hernantes, Gorka Gallardo, and Nicolás Serrano. It infrastructure-monitoring tools. *IEEE Software*, 32(4):88–93, 2015. doi:10.1109/MS.2015.96.
- [2] Oracle. Oracle real application clusters 19c technical architecture. URL: <https://www.oracle.com/webfolder/technetwork/tutorials/architecture-diagrams/19/rac/main.html>.
- [3] Vadim Ipatov. Developing plugins for zabbix agent 2, Jul 2020. URL: <https://blog.zabbix.com/developing-plugins-for-zabbix-agent-2/9682/>.
- [4] Don Jones. Creating unified it monitoring and management in your environment. *Jones–San Francisco: Realtime Publishers*, 2012.
- [5] Splunk. What is it monitoring? URL: https://www.splunk.com/en_us/data-insider/what-is-it-monitoring.html#it-monitoring-overview.
- [6] IBM. Database monitoring, Aug 2021. URL: <https://www.ibm.com/docs/en/db2woc?topic=reference-database-monitoring>.
- [7] Oracle. Database. URL: <https://www.oracle.com/pt/database/>.
- [8] David Baum. Oracle enterprise manager. Technical report, Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA, 2020. URL: <https://www.oracle.com/pt/a/ocom/docs/enterprise-manager/oracle-enterprise-manager-solution-brief.pdf>.
- [9] Zabbix. *Zabbix 5.4 Manual*. Zabbix. URL: <https://www.zabbix.com/documentation/current/manual>.
- [10] Zabbix. *Zabbix 6.0 Manual*. Zabbix. URL: <https://www.zabbix.com/documentation/current/manual>.
- [11] Oracle. What is a relational database (rdbms)? URL: <https://www.oracle.com/pt/database/what-is-a-relational-database/>.
- [12] Oracle. Oracle instant client, 2018. URL: <https://www.oracle.com/database/technologies/instant-client.html>.
- [13] Ubuntu. Oracle instant client, 2015. URL: <https://help.ubuntu.com/community/Oracle%20Instant%20Client>.