Description of the changes compared to standard plugin.

E-Mails:
Manfred Comes <mcomes@gmx.net>
Eduardo Willame (Valentim) <eduardowillame@yahoo.com.br>

Overview

Compared to standard plugin `check_oracle` functionality was extendet. It is now possible to check autoextend tablespaces by using the new Parameter `--tablespace_AE` (added by Manfred Comes) and the functionalty of parameter `--tablespace` was extendet (by Eduardo Williame), so that it is now possible to check temporary tablespaces with this parameter (of cause it is still possible to check permanent tablespaces).

Parameter `--tablespace` don't check against the maximum autoextend size. It checks against the free blocks in the datafiles.

Parameter `--tablespace_AE` can't check temporary tablespaces.

For performace Issues bind variables are used.

Eduardo Williams:
When Nagios call the plugin, this executes the same query for all tablespaces at the same time, but for oracle this does not the same query generating different SQL_IDs for each one. With binds is easy to get the plan and improve it through tools like EM or other one, besides to consume less CPU resources.

Autoexetend tablespaces, parameter `--tablespace_AE`

By calling `./check_oracle_v3 --help` you see:

```
--tablespace_AE
   For tablespaces with autoextend manangement !!!

   Check local database for tablespace capacity in ORACLE_SID
       --->  Requires Oracle user/password specified.
                --->  Requires select on dba_data_files and dba_free_space

Explanation of parameter --tablespace_AE
========================================
IT checks against maximum autoextend size if tablespace datafile(s) is/are in
automatically extend modus. Be aware that there has to be enough diskspace,
this plugin doesn't check that!

If automatically extend modus is for all datafiles of the tablespace not set,
this paramter yields the same result as --tablespace :-)

This paramteter don't work with temporary Tablespaces, use --tablepspace for that !
```

The query of the standard plugin was extended, this had to be done in to Stepps:

First step was, not to do the aggragation over the datafiles inside of the inner select statement,. This aggregation is now done in the outer select statment. By doing so, it gets possible to compare the data of `dba_data_files.maxbytes` (which ist the possible size of autoextension) and `dba_free_space.bytes` (which are the allocated but not uses bytes). Therefor it is nessary to aggregate alle values `dba_free_space.bytes` per datafile, because there may be more than one entry per datafile.

How does it works?

If no autextend is set, parameter `--tablespace_AE` yields the same values than parameter `–tablespace` which does exactly the same as in the standard pluging.

When autextend is not set for the datafile `dba_data_files.maxbytes` is `0` or Null.

Limitations:
The script calculates the follwing maximum over all datafiles building the sum in the end:

```
greatest(a.AE_free, b.free)
```

It is

```
a.AE_free = (NVL(dba_data_files.maxbytes,0.0) - dba_data_files.bytes) / 1024 / 1024
```

which ist the space in MByte that is not used by extends up to now.

On the other hand

```
b.free
```

yields by

```
sum(NVL(dba_free_space.bytes, 0.0))/1024/1024
```

(the sum is calculated for all free blocks per datafile)

It therefore is the caluculated size of the free blocks in the datafile in MByte

It would be better to:

- If autoextend is set to calculate the sum of both values.
- If autoextend is not set only to take `b.free`

In other words if autoextend is set the calculeted value is a little bit to low. For warning issues this seams not as problem to me. Doing better would be possible by using PL/SQL.