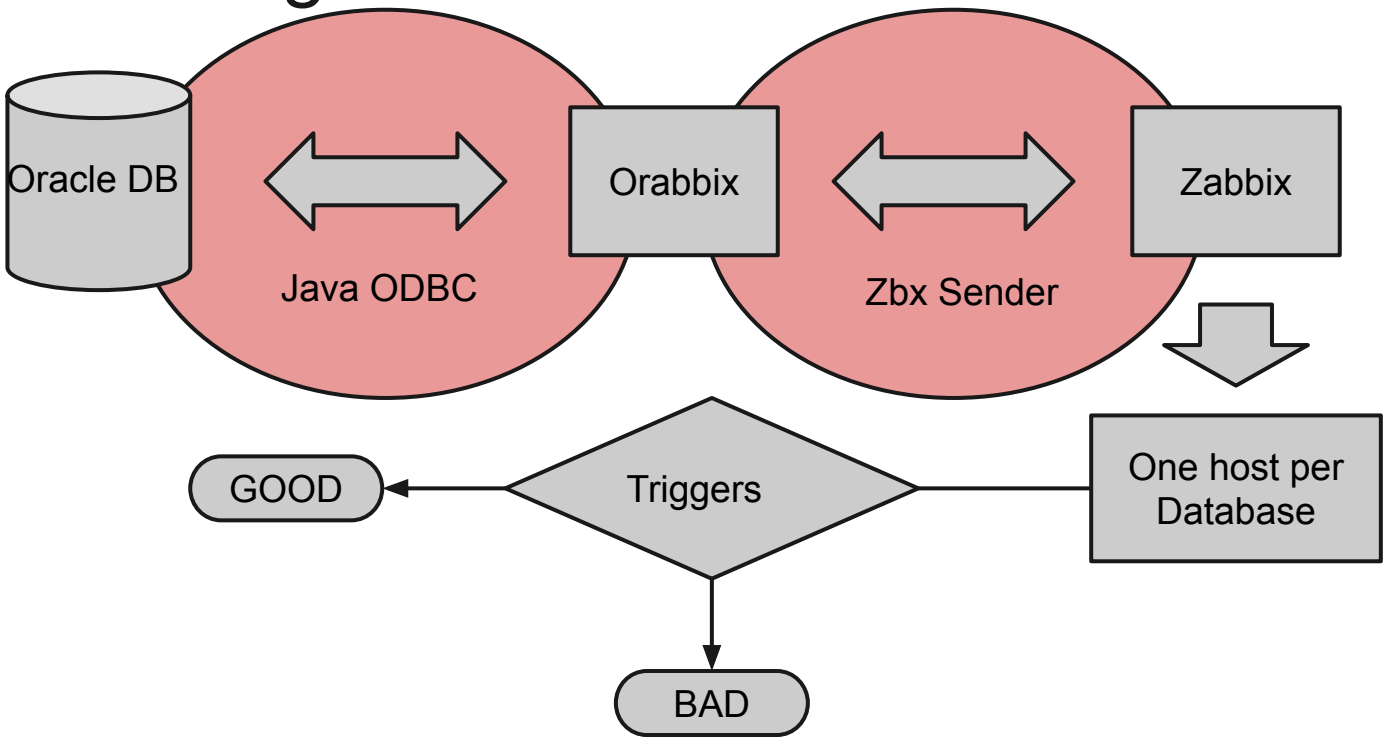
Perobbix+Zabbix DB Monitoring

Advanced techniques for DB Monitoring

Development by Julio C.Hegedus
Oracle Experience from Felix Flux and Ramon Bouthoorn

The past with Orabbix

Orabbix is a java daemon that runs a list of queries against databases.



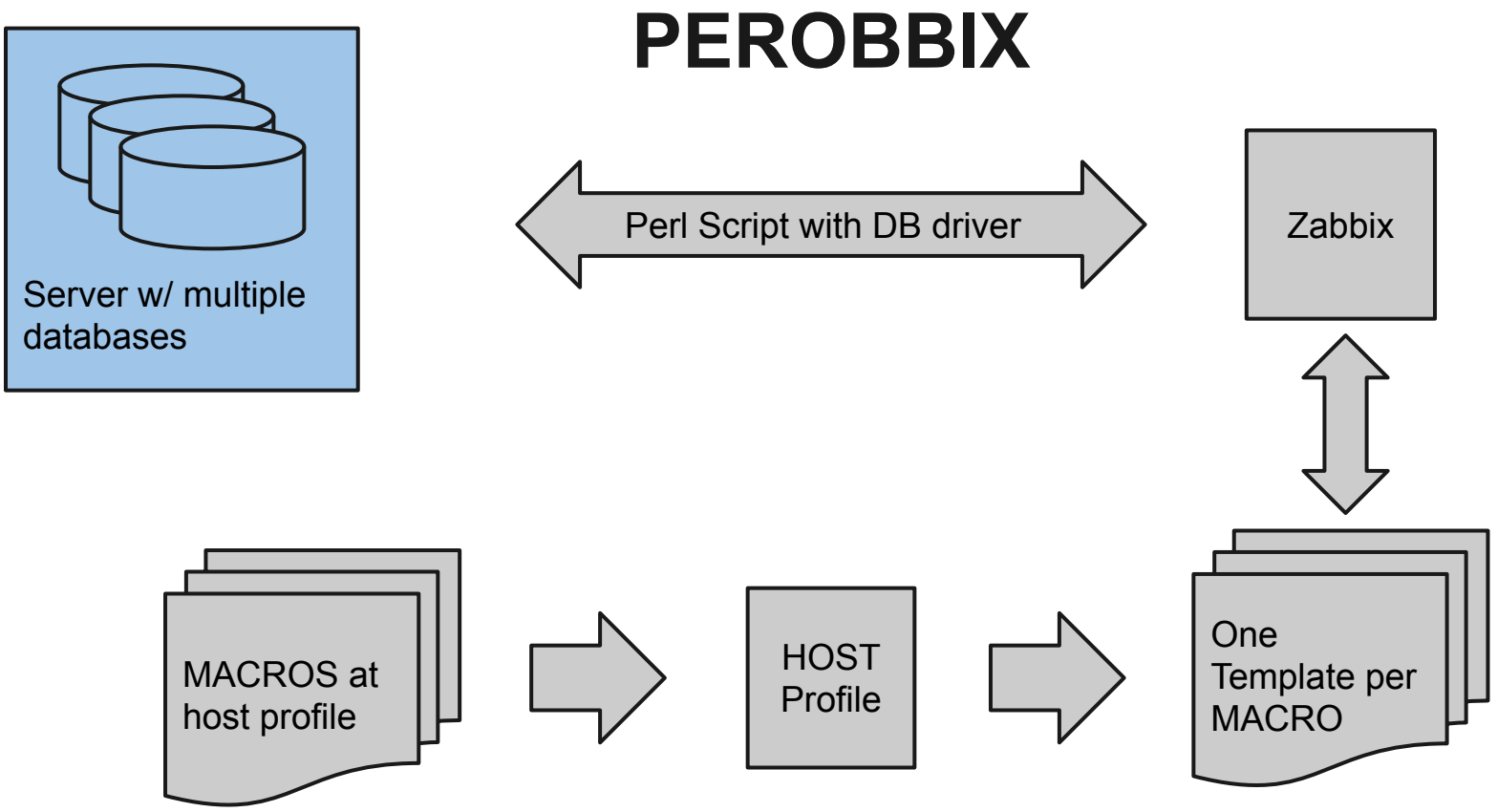
What we are missing?

- When the Java daemon misbehaves we may miss information;
- When Zabbix is busy (i.e.: doing backup) zabbix_sender struggles;
- No timing information whatsoever;
- Multiple hosts for multiple DBs in one server is confusing;
- The program only allowed Oracle DBs to be seen.

A New Model

- Zabbix should query the databases by itself, therefore an external command;
- All DB from a host in one profile;
- Proficient time measurements;
- Flexible model of connection, possible to different database flavors;
- Easy frontend only install / deletion of databases;
- Reliability, multiplicity, easy maintenance and local development of new features.

Perl + Oracle + Zabbix



The Perl Script Method

- Current version for Oracle runs in under 700 lines (including comments);
- Can use different ODBC methods for connecting with DBs;
- High Resolution timer for each single action;
- Easy maintenance and easy testing;
- Manual debugging as a regular script;
- Fetch all errors with accuracy;
- Timeout controllable and variable;
- Low security concerns (no pwds shown).

Zabbix side alterations

- (Source Patch) Default timeout from 30 to 300 seconds at Server and Proxy binaries;
- (Source Patch) Disable double quoting and parameter escaping (introduced in version 2);
- Maybe modules from Perl or OS need to be installed;

Server (DB) side alterations

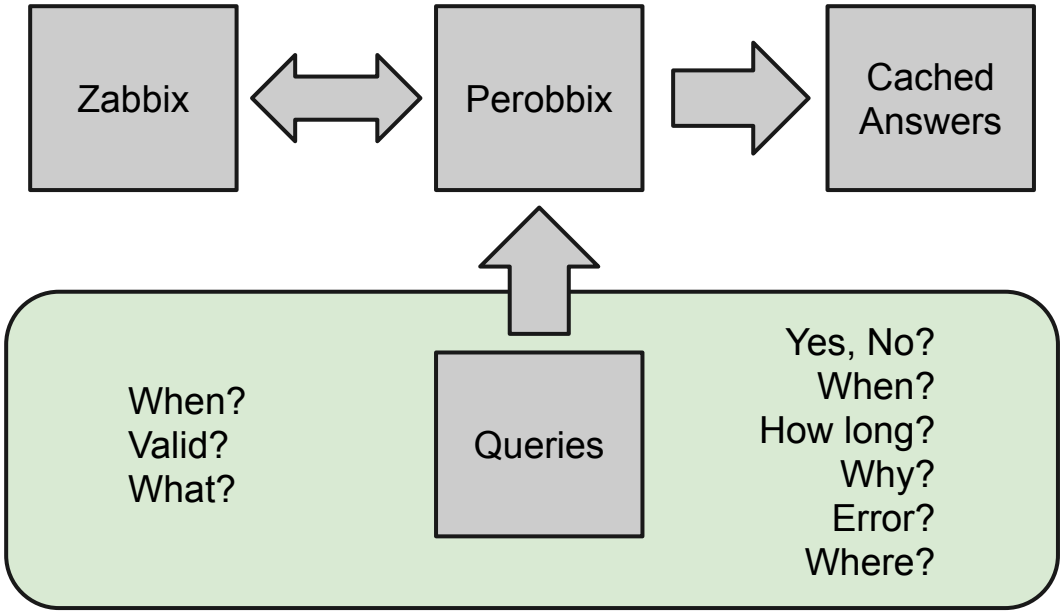
- Requires a user RO (at least) capable of reading and executing the proposed queries against the databases to be tested;
- Required to receive connections via network;
- No zabbix_agent is required on the server
- Sometimes other methods used in Agentd userparameter.

Server | Host Profile

- Regular information;
- Attach to Template_DB_NAME
- Add MACRO like:
  - {$DB_NAME} => DBNAMEVAL

Methodology (1/2)

A Batch Check, Cached responses, Timely execution



Methodology (2/2)

Fetch values you need to evaluate situation

A Template

Items to be executed

Name: Oracle {$DB_NAME} Batch Check

Batch: perobbix.pl[{HOST.CONN} -P passwd -u user -D {$DB_NAME} -q query.file -zs]

Key Val 1: zstatoracle[{HOST.CONN} -d {$DB_NAME} -q query.file -o logincount -t]

Key Val 2: zstatoracle[{HOST.CONN} -d {$DB_NAME} -q query.file -o logincount]

A Query File

(control)QueryList=logincount

(3)logincount.Query=SELECT ...
(2)logincount.RaceConditionValue=TRUE
(1)logincount.RaceConditionQuery=SELECT ...
(0)logincount.Period=5
(4)logincount.NoDataFound=0

A Result File

1x.xx.x.x-DBNAME-query.file-user-rfile.log

logincount=3;0.001852;1367500919
totalexec=0.582413;1367501218;1367501218
errorexec=none
errorconn=no
conntime=0.065326
dumpcomplete=yes

Timings & Values

- Perobbix provides overall batch check time;
- Connection between Zabbix <-> Database measured;
- Each query have its one time counter;
- Overall run must go before zabbix timeout, so each counter decreases total time limit left - catch timeout exit better than die;
- Every single error (if so) is raised;
- Last execution of everything is kept;
- Every query has its own timer;

Hummm and now what?

- Reverse MACRO translation does not work.

When sending a DBNAME that should match a value from a MACRO, Zabbix_Sender and server are unable to identify which Key it is. Only work if zabbix_sender uses the MACRO name to refer to a Key (same syntax the key is written, i.e: {$DB_NAME})

Closing Comments

- Triggering;
- Notification richness;
- Wealth of information;
- Testing and much more with the data;
- Statistical analysis;
- Correlation of factors;
- Graphing;
- Error debugging tool;
- Flexibility and freedom;