



PALOMINO DB

Proven Database Excellence

Nagios Plugins for Monitoring MySQL

Presented by:

Sheeri K. Cabral

Database Operations Manager

www.palominodb.com

What is Monitoring?

- Threshold alerting
- Graphing/trending



Why Monitor?

- Problem alerting
- Find patterns
- Early warning for potential issues



Open Source Alerting Solutions

- Nagios
- Cacti
- Munin
- Zabbix
- Zenoss
- OpenNMS



Commercial Alerting Solutions

- MySQL Enterprise Monitor
- MONyog
- Oracle Grid Control
- Hyperic HQ



Best to use what you already have



Nagios is great because...

...anyone can write a plugin



The problem with Nagios...

...anyone can write a plugin



Official Nagios Plugins for MySQL

- `check_mysql`
- `check_mysql_query`



check_mysql

- db connectivity
- slave running
- slave lag using seconds_behind_master

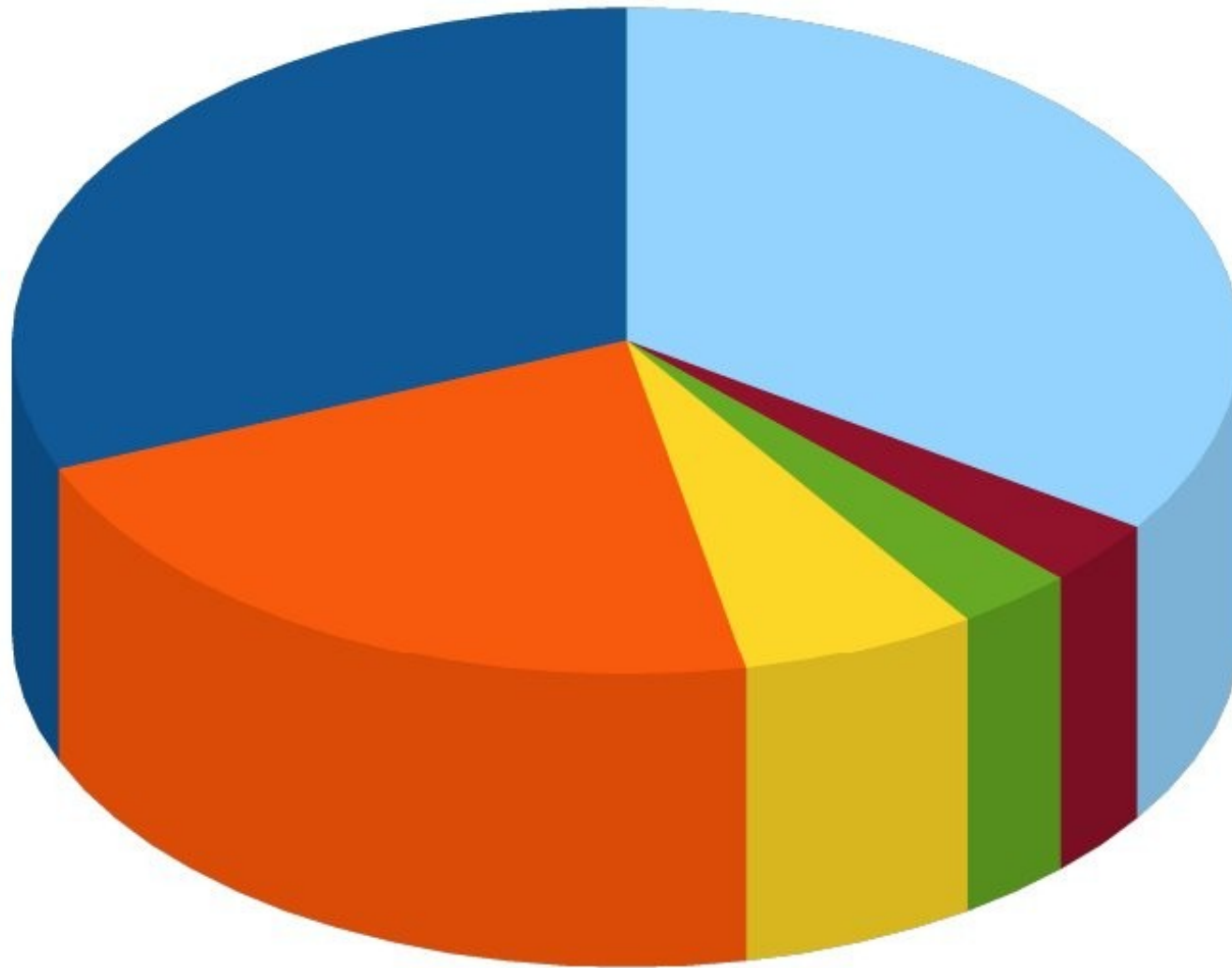


check_mysql_query

- Checks the output of a query is within a certain range (numerical)



MySQL Plugins on Nagios Exchange



■ slave ■ variables ■ # connections
■ deprecated ■ MSSQL ■ Other



System vars**Status vars****Caching****Calculations**

| | | | | |
|----------------------|-----|------|-----|------------|
| Check 1 | yes | yes | no | no |
| check_mysql | no | yes | no | no |
| check_mysqlid_status | no | yes | no | no |
| check_mysql_stats | yes | no | yes | no |
| check_mysqlid | no | many | no | no |
| check_mysql_health | yes | yes | no | Hard-coded |
| Check 7 | yes | yes | no | Hard-coded |



“Let us know what you'd like to see!”



What I want

| | System vars | Status vars | Caching | Calculations |
|------------------|--------------------|--------------------|----------------|---------------------|
| New check | yes | yes | yes | flexible |



Caching

- Save information to a file
 - `--cache-dir /path/to/dir/`
 - Use the file instead of connecting again
- `--max-cache-age <seconds>`
- `--no-cache` to force connection



--mode=varcomp

- %metadata{varstatus}
- SHOW GLOBAL VARIABLES
- SHOW GLOBAL STATUS
- --expression allows word replacement
- --comparison flexible



Sample Command Definition

```
define command {  
    command_name      check_mysql_tmp_tables  
    command_line      $USER1$/mysql_health_check.pl  
    --hostname $HOSTADDRESS$ --user myuser --password mypass  
    --cache-dir=/var/lib/nagios/mysql_cache  
    --max-cache-age=300  
    --mode=varcomp  
    --expression=  
        "Created_tmp_tables/Uptime_since_flush_status"  
    --comparison=">(10/60) "  
}
```



Sample Service Definition

```
define service {  
    use                generic-service  
    host_name          __HOSTNAME__  
    service_description MySQL Temp Tables  
    check_command      check_mysql_tmp_tables  
}
```



--mode=long-query
--mode=locked-query

- %metadata{proc_list}
- SHOW FULL PROCESSLIST



Extending Information

```
sub fetch_server_meta_data {}
```

add a new hash key to %meta_data

```
$meta_data{innodb_status} =  
$dbh->selectall_arrayref(  
q|SHOW ENGINE INNODB STATUS|);
```



Currently Unused

- `%metadata{innodb_status}`
 - `SHOW ENGINE INNODB STATUS`
- `%metadata{master_status}`
 - `SHOW MASTER STATUS`
- `%metadata{slave_status}`
 - `SHOW SLAVE STATUS`



Limitations

- --comparison for critical only, not warning (in progress)
- Cannot compare to previous (in progress)
- One check/calculation per Nagios service
- Does not output for performance data



Questions? Comments?

