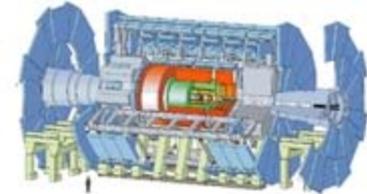




WLCG 3D Workshop April 2009



the ATLAS Experiment

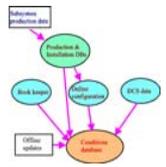


Preventing Oracle Overload with Pilot Query

Florbela Viegas



Presentation Summary



- Introduction
- What is overload?
- The ATLAS COOL Pilot solution
- Tests at Lyon - Lessons learned
- A smarter Pilot - version 2
- Future tests and plans



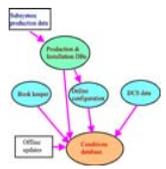
What is overload?



- Main concern in access to the Tier-1 (and CERN) databases is avoiding overload of resources.
- Especially important because when databases have resource bottlenecks Oracle Streams replication is impaired and measures have to be taken at CERN to avoid consequences to the other sites.
- So what is this « overload » we have to avoid?
- At database level, overload happens when resources are low and bottlenecks appear for:
 - **CPU** - when too many active processes cause the CPU to go near 100%.
 - **Disk** - writing activity is especially damaging to the Streams replication - one incident occurred with TAG uploading. Reading activity competes with writing, and can indicate performance problems in COOL.
- Oracle provides metrics that can be used to quantify this load:
 - Enterprise Manager shows the accumulation of active sessions on events for a given period of time, in a graphical form - main form of evaluation of "overload" for DBAs that use this tool.
 - "OS Load" metric (updated every minute) shows the number of processes that are either running or in the ready state, waiting to be selected by the operating-system scheduler to run.



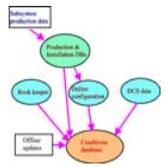
The ATLAS COOL Pilot solution



- To avoid overload in the Oracle Database, we need to limit the number of jobs that access the database, according to a criteria of load.
- At database level, the option for limiting number of sessions is simplistic, as it does not take into account the status of each one and overall database activity.
- So a new mechanism is needed that throttles the submission of jobs to the database, using feedback from the database activity to determine if jobs can access, or have to wait for a better time.
- Sasha Vaniachine suggested the creation of an « ATLAS COOL Pilot » query, that would be added to all the jobs, and used to throttle the access to the database, before the job started its work.
- This pilot, a first simple version, was implemented and tested at IN2P3, which held very interesting results.



Tests at Lyon - Setup



- This code was added to the job:

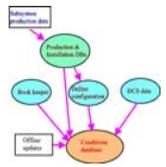
```
"""File: pilot.py
....
while True :
....
    sensor.execute("select atlas_cool_pilot.load_status from dual")
....
    if status=='GO' :
        # RW: avoid start of many jobs released at once
        sleep (attempt*random.randint(1, 300))
        #exit(134)
        break
    else:
        print strftime(" %a, %d %b %Y %H:%M:%S ", gmtime()), " avoiding load of", load[3],
            "at", sessions, "concurrent COOL sessions"

        if attempt<5 :
            attempt=attempt+1
            print attempt*sessions
            sleep (attempt*sessions)
....
    if attempt==5 :
        print " FATAL: Killing job to avoid Oracle overload"
        sys.exit(134)
....
```

- The pilot code in PL/SQL returns the status « GO » or « NOGO » according to load.



Tests at Lyon-Lessons learned



- The jobs were throttled using different threshold values, and this code seemed to work well:

Sample output from the finished job:

running on ccwl0613 on Mon Mar 2 13:06:01 2009

Database operations pilot at LYON

pilot detected status GO - Load: 03.10 Sessions: 813 Threshold:4

An example of a job held at a lower threshold:

Database operations pilot at LYON

pilot detected status NOGO - Load: 03.00 Sessions: 408 Threshold:2

Mon, 02 Mar 2009 12:21:35 avoiding load of 03.00 at 408 concurrent COOL sessions

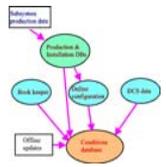
pilot detected status NOGO - Load: 02.10 Sessions: 477 Threshold:2

Mon, 02 Mar 2009 12:28:23 avoiding load of 02.10 at 477 concurrent COOL sessions

pilot detected status NOGO - Load: 02.70 Sessions: 483 Threshold:2



Tests at Lyon-Lessons learned



- The jobs that failed due to timeout, returned a special code that scheduled for resubmission at a later time:

- ATLAS pilot query approach can kill jobs that waited too long:

Database operations pilot at LYON

pilot detected status NOGO - Load: 02.80 Sessions: 420 Threshold:2
 Mon, 02 Mar 2009 12:21:26 avoiding load of 02.80 at 420 concurrent COOL sessions
 pilot detected status NOGO - Load: 02.40 Sessions: 471 Threshold:2
 Mon, 02 Mar 2009 12:28:26 avoiding load of 02.40 at 471 concurrent COOL sessions
 pilot detected status NOGO - Load: 02.90 Sessions: 776 Threshold:2
 Mon, 02 Mar 2009 13:08:27 avoiding load of 02.90 at 776 concurrent COOL sessions
 pilot detected status NOGO - Load: 03.40 Sessions: 701 Threshold:2
 Mon, 02 Mar 2009 14:00:11 avoiding load of 03.40 at 701 concurrent COOL sessions
 FATAL: Killing job to avoid Oracle overload

Panda job information

Jobs: 27701616

[Click for help](#)

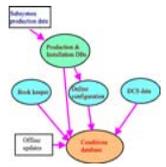
Showing 1 jobs modified from 2009-03-02 14:59:36 to 2009-03-02 14:59:36

Jobs:

PandaID, Owner	Job	Status	Created	Time to start	Duration	Ended/ Modified	Cloud/Site, Type
27791616 Rodney.Walker@physik.uni-muenchen.de	data08_cvalid.00088425_physics_MBT5_BCM_LUCID.recon.o4_r625_tid042225_00177.job #1	failed	03-02 10:30	0:37:54	3:50:43	03-02 14:59	FR/LYON, production
Error details: trans: Athena core dump or timeout, or conddb DB connect exception In: data08_cosmag.00088425_physics_MBT5_BCM_LUCID.daq_RAW.o4 Out: data08_cvalid.00088425_physics_MBT5_BCM_LUCID.recon.HIST.o4_r625_tid042225							



A smarter pilot – version 2



- Still, this code does not account for surges of jobs, as happened accidentally at Lyon, as it does not control the number of jobs submitted.
- Also it is difficult to analyze job behaviour because only the job logs have the messages, so no telling if the job is waiting because of pilot query or of other factors.
- So, a smarter pilot is required that:
 - Knows how many jobs can be submitted according to the measure of load
 - Controls and holds the jobs until capacities exist
 - Works like a common ticketing system
 - Keeps track of workflow of jobs, and logs behaviour
- So version 2 is developed and being conceptually tested now:
 - It keeps track of requests
 - Jobs only advance when they are « called »
 - Records request time and service times
 - Allows for post-mortem analysis and simulation of job behaviour



A smarter pilot – version 2



- New pilot concept and code:

TS_PARAMETERS

NAME	VALUE
RUN	1
LOAD_THRESHOLD	2
NUM_TICKETS_UNIT_LOAD	1
SCHEDULER_WAIT_SECONDS	30
CLIENT_WAIT_SECONDS	45
TICKET_HANDLER_JOB	3799598
LOAD_FUNCTION	RAC_OSLOAD

TS_BOARD

TICKETNO	TAKEN	SERVICE_INFO	TS_CALLED	TS_ATTENDED
8	1	ATLAS_COOL_READER_U experiment-- Requested Service at : 03.mar.09 10:09:13-- Called at : 03.mar.09 10:09:43-- Attended at : 03.mar.09 10:09:58	03-Mar-2009 10:09:43 AM	03-Mar-2009 10:09:58 AM

TS_REQUEST

TICKETNO	SERVICED	CLIENT_INFO	TS_REQUEST	TS_CALLED	TS_ATTENDED	TS_LEFT
8	1	ATLAS_COOL_READER_U experiment-- Requested Service at : 03.mar.09 10:09:13-- Called at : 03.mar.09 10:09:43	03-Mar-2009 10:09:13 AM	03-Mar-2009 10:09:43 AM	03-Mar-2009 10:09:58 AM	



Pilot Request Function
(Get ticket,wait for turn,
wake up when called or timeout)

Get requests, check load,
call tickets according to load
brackets, go to sleep.



Conclusions and Future plans



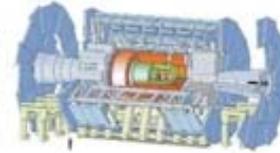
- Determine measure of load that is best representation, with subsequent interval of time
- Determine the best holding pattern (client wait vs database sleep). Event generated in database is : Wait (Idle, « PL/SQL lock timer ») so it will not register in the blocking alarms, but how will the batch system interpret it?
- Set tests at each Tier-1 to determine best values for parameters, especially number of jobs per unit of load to release.
- Results were promising but tests have to continue.
- More information in:
 - COOL reprocessing Twiki: [CoolReprocessingTests](#)
 - CVS Code: <http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/groups/Database/CoolPilot/>
 - New site being developed by Elizabeth Vinek for overview of activity: <http://test-cool-pilot-query.web.cern.ch/test-cool-pilot-query/>



Dashboard



the ATLAS Experiment



ATLAS COOL Pilot Query Monitoring Page

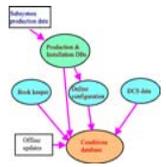
Information about the Pilot Query: [CoolReprocessingTests TWIKI](#)

[Stats/Graphs](#) [Animation](#)

Overview		Min, Max, Mean	
# tickets waiting	0	Mean time requested-called in last 24h	no info
# tickets completed	15	Min time requested-called in last 24h	no info
# tickets completed in last 24h	0	Max time requested-called in last 24h	no info
List of tickets waiting		Mean time requested-called last 10 tickets	283
List of tickets completed		Min time requested-called last 10 tickets	28
		Max time requested-called last 10 tickets	2430
Clients that left		Current DB load	
Clients having left (in total)	0	load as in gv\$osstat .589	
Clients having left in the last 24h	0		
List of clients having left			



More information

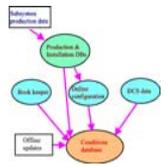


- In2P3 database status at time of Session maxed out problem:

Oracle Enterprise Manager 10g
Grid Control
Home Targets Deployments Alerts Compliance Jobs Reports
Hosts[Databases]Application Servers[Web Applications]Services[Systems]Groups[All Targets]
Cluster: frocin2p3_ors_atlas > Cluster Database: dbatl.in2p3.fr > Database Instance: dbatl.in2p3.fr_dbat14 > Logged in As STRMADMIN
Top Activity
Switch Database Instance dbatl.in2p3.fr_dbat14 Go
View Data Historical
Active Sessions
6:00 8 10 12 PM 2 4 6 8 10 12 AM 2 4
Detail for Selected 30 Minute Interval
Start Time 03-Mar-2009 16:30:34 o'clock CET KPI ASH Report
Top SQL
Schedule SQL Tuning Advisor Create SQL Tuning Set
Select All Select None
Select Activity (%) SQL Hash Value SQL Type
95.12 d5jx122377a03 SELECT
4.88 4qd6b1r53vt88 UNKNOWN
select atlas_cool_pilot_load_status from dual
Top Sessions
View Top Sessions
Activity (%) Session ID User Name Program
59.6 214 SYS oracle@ccdbcl13 (m000)
1.32 306 SYS oracle@ccdbcl13 (QMNC)
1.32 319 SYS oracle@ccdbcl13 (MMON)
0.66 205 SYS
0.66 86 SYS
Eric Lancon
Re: monitoring
Merci!
On 05 Mar 2009, at 15:38, Florent Tiqué/Aires Viegas (TNS-V1-V3)
ATLAS COOL READER II
Local intranet 90%
19/04/2008



More information



- In2P3 database status at time of Session maxed out problem:

Oracle Enterprise Manager (FJRIEGASOEM) - SQL Details: d5jx122377a03 - Windows Internet Explorer provided by CERN

http://oms3d.cern.ch:4889/em/console/database/instance/sqlDetail?event=doLoad&target=dbatl.in2p3.fr_dbatl4&type=or...

T1 Statistics Oracle Enterprise Manag... 3D meeting (05 March 2009)

Details

Select the plan hash value to see the details below. Plan Hash Value: 1388734953

Statistics Activity Plan Tuning Information

Summary

Drag the shaded box to change the time period for the detail section below.

Detail for Selected 30 Minute Interval

Start Time: 03-Mar-2009 16:30:34 RU1 ASH Report

Activity (%)	SID	User	Program	Service	Plan Hash Value
2.22	214	ATLAS_COOL_READER_U	python@ccw10648 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	86	ATLAS_COOL_READER_U	python@ccw11407 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	214	ATLAS_COOL_READER_U	python@ccw10653 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	205	ATLAS_COOL_READER_U	python@ccw11435 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	205	ATLAS_COOL_READER_U	python@ccw11042 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	86	ATLAS_COOL_READER_U	python@ccw11145 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	86	ATLAS_COOL_READER_U	python@ccw11443 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	214	ATLAS_COOL_READER_U	python@ccw10499 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	214	ATLAS_COOL_READER_U	python@ccw11155 (TNS V1-V3)	dbatl.in2p3.fr	1388734953
2.22	205	ATLAS_COOL_READER_U	python@ccw10597 (TNS V1-V3)	dbatl.in2p3.fr	1388734953

Local intranet 90% 4:16 PM