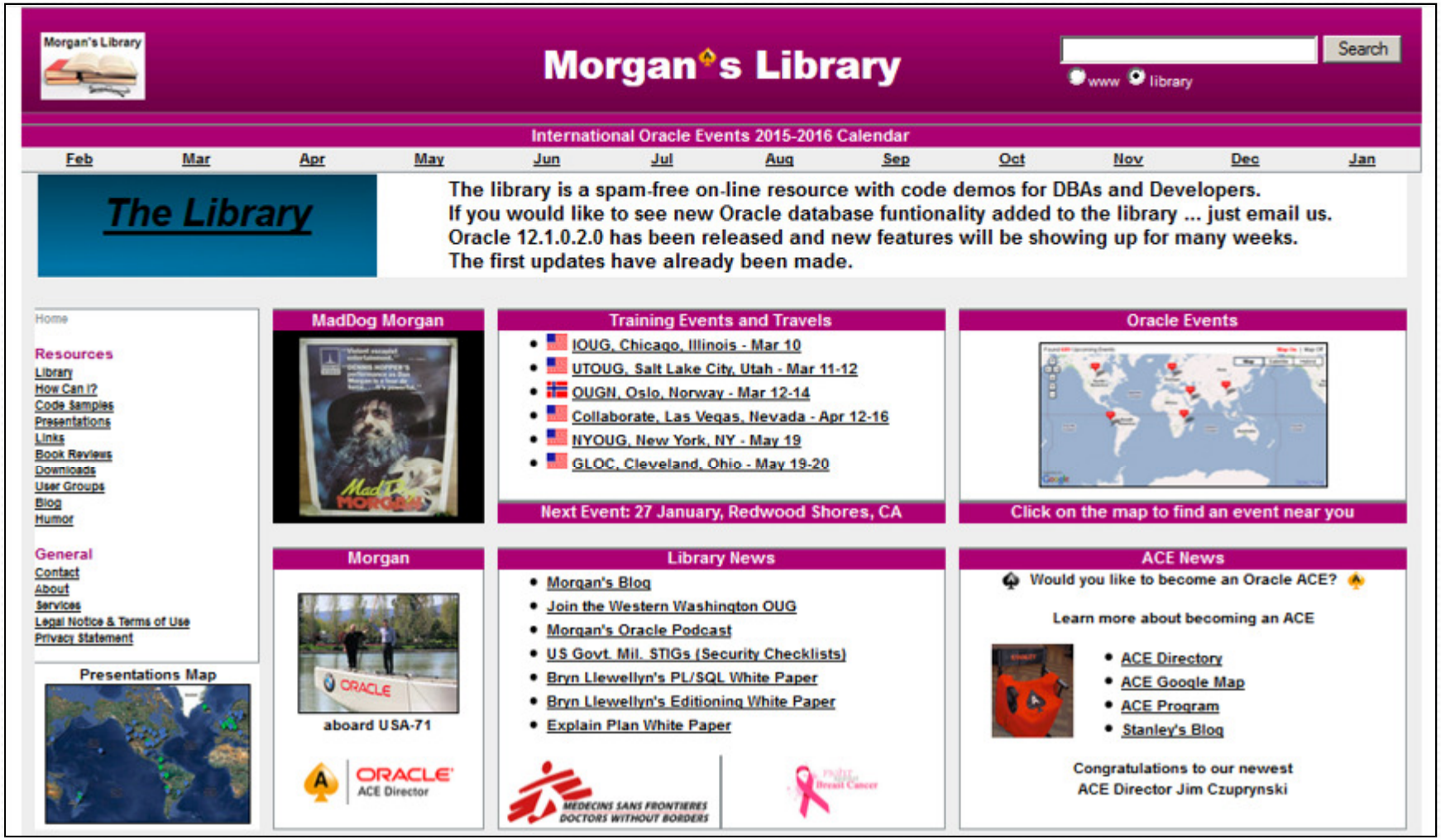
oRCAle World: Root Cause Analysis

Daniel A. Morgan
Forsythe Meta7

Daniel A. Morgan

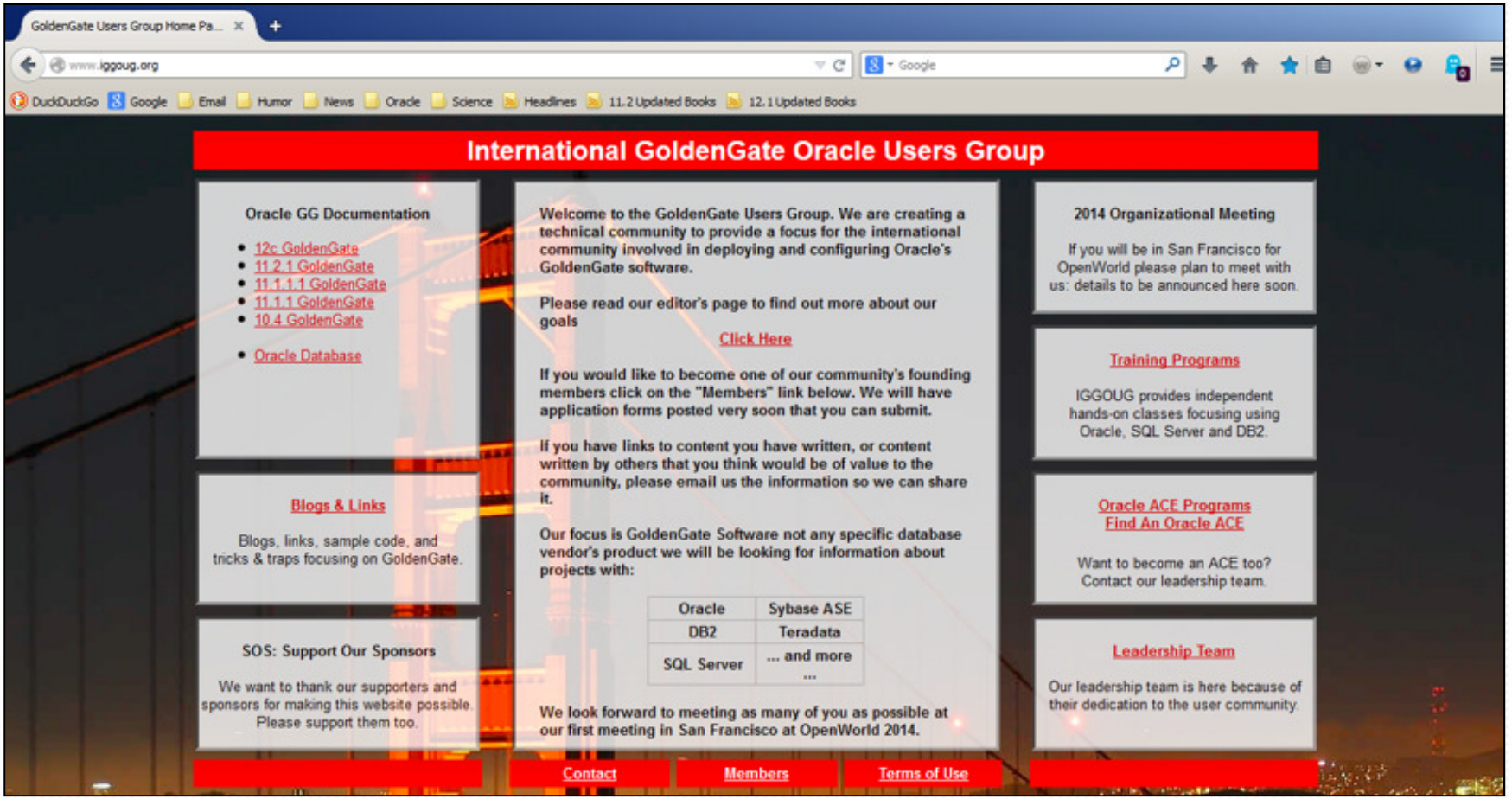
- Oracle ACE Director
- Educator
  - Author Oracle curriculum and primary program instructor at University of Washington
  - Oracle Consultant: Harvard University
    - University Guest Lecturers
      - APAC: University of Canterbury (NZ)
      - EMEA: University of Oslo (Norway)
      - Latin America: Universidad Latina de Panama and Technologico de Costa Rica
- The Morgan behind Morgan's Library on the web
  www.morganslibrary.org
- 10g, 11g, and 12c Beta tester
- Co-Founder International GoldenGate Oracle Users Group
- Principal Advisor: Forsythe Meta7

The Morgan's Library Web Site



www.morganslibrary.org

IGGOUG: The New Users Group In The Neighborhood



www.iggoug.org

Forsythe Meta7

ORACLE Platinum Partner

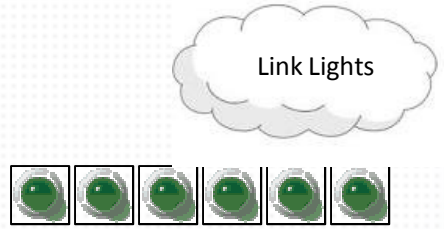
- Meta7 is an Oracle-exclusive division of Forsythe dedicated to supporting the Oracle's technology stack
- Our practice is highly focused and supports
  - Oracle's full range of products
    - Infrastructure (Exadata, Exalogic, Storage, Network)
    - Software (Database, GoldenGate)
  - Oracle's full range of services focusing on Cloud
  - Meta7's internally developed processes and procedures optimizing
    - Deployment, Configuration, and Integration
    - Migrations and Upgrades
    - Performance Tuning
    - End-to-end Security
  - Knowledge Transfer
  - ... and, of course, **R**oot **C**ause **A**nalysis and troubleshooting

META7™ Solutions for the Red Stack

Travel Log: OTN LAD Tour 2014

First Principles

- If the database is unavailable it is a database problem
- If the database is slow it is a database problem
- Oracle DBAs are expected to find database problems
- To fix the database so the problem never happens again
- To write an RCA (Root Cause Analysis) document
- Which will point to the database
- Even when the database isn't the root cause
- Because that is what DBAs are trained to do
- Did I mention ... the network is just fine?



My Blog: January 10, 2011

If databases were run with the same degree of intelligence and attentiveness as a network router we would:

log in as SYS, type
`SELECT * FROM dual;`
and if we did not get an exception, declare everything was fine.

[network admins]..., I might be inclined to recommend that we plug a couple of them into their own networks and see if they light up.

What Is Root Cause Analysis?

- Root Cause Analysis is finding, fixing, and reporting on, the event that precipitated a service related incident
- The incident or change may have resulted in one or more failures that affected database performance and/or availability
- Sometimes the root cause is within the database, for example a bug, it is at least equally probable the database was an innocent bystander

- The point is to avoid repeated fire fighting exercises

Root Cause Analysis Doc (1:7)

Root-Cause Analysis - PROVISIONAL

Node 5 Eviction 2010-12-06 18:32

Author:	Daniel Morgan
Date:	8 Dec 2010
Version:	v 1.0
Status:	Provisional
Reference:	RCA-PROV_201012061832_Node5
Sec. Class:	Commercial in Confidence

Root Cause Analysis Doc (2:7)

Contents

Root Cause Analysis Doc (3:7)

1 Introduction

1.1 Scope

This document is a root-cause analysis for the OPTIMA OPM01P database event that started at on Monday Dec 06 at 18:32:12 2010 on instance opm01p5. It is current a PROVISIONAL assessment and subject to change

1.2 Revision History

Revision Number	Date	Name	Revision
1.0a	8 Dec 2010	Daniel Morgan	Initial Version of Document

1.3 Review/Approval

Reviewer	Date	Feedback
S Bowke	8 Dec 2010	Approved for release as a PROVISIONAL root-cause assessment for discussion purposes

1.4 Outstanding Actions List

This section is to be used to capture the outstanding actions

Outstanding Actions List					
Action No	Priority	Owner	Description	Date Raised	Status
1	1	Daniel Morgan	Issue Full RCA	8 Dec 2010	In-Progress

Root Cause Analysis Doc (4:7)

2 RCA Timeline

At 6.32 Dec 6, Node opm01p5 alert log makes reference to the global resource directory, the global resource directory keeps details of how resources are used within a cluster database.

Mon Dec 06 18:32:12 2010
Reconfiguration started (old inc 16, new inc 18)
List of nodes:
4 5 6 7
Global Resource Directory frozen
Communication channels reestablished
Master broadcasted resource hash value bitmaps
Non-local Process blocks cleaned out
Mon Dec 06 18:32:21 2010
LMS 1: 33 GCS shadows cancelled, 1 closed, 0 Xw survived
Mon Dec 06 18:32:21 2010
Mon Dec 06 18:32:21 2010
LMS 0: 1 GCS shadows cancelled, 0 closed, 0 Xw survived
LMS 2: 2 GCS shadows cancelled, 0 closed, 0 Xw survived
Mon Dec 06 18:32:23 2010
Set master node info
Submitted all remote-enqueue requests

The LMON was unable to fix the issue and subsequently crashed at 6.37 (several related bugs apparent on metalink)

Mon Dec 06 18:37:12 2010
LMS2 (ospid: 13489) has not called a wait for sub 0 secs.
ERROR: GES receivers are not healthy.
ERROR: LMON (ospid: 13477) is terminating the instance.
ERROR: Please check LMON trace file for more detail.
LMON (ospid: 13477): terminating the instance due to error 481

This in turn caused the instance to restart

USER (ospid: 9864): terminating the instance
Instance terminated by USER, pid = 9864
Mon Dec 06 18:37:47 2010
Starting ORACLE instance (normal)

Root Cause Analysis Doc (5:7)

The instance then was unable to open as the global resource directory started to relocate the Global objects (this can take several minutes / hours).

DBAs, Oracle and System Administrators worked on the outage. DBA restarted Node 5 several times throughout the night however each time it came up queries to the GV$views were failing on the system.

At 7:30pm Dec 6 XXX Recommended a full Database bounce to restore the System

At 8:39 PM /var/adm/messages on usp9004a (opm01p5) had the following entry:

```
Dec  6 20:39:51 usp9004a ip: [ID 448101 kern.error] nxge5: <unknown
primitive> failed: DL_UNSUPPORTED
```

At 05:24am Dec 7 a secondary problem occurred on the system as /appl/oracle on usp9004b (opm01p6) was allowed to be filled up with logs and trace files being collected on the system.

Full Database shutdown was executed at 6:50am server time on Dec 7th.
OPTIM_REPORT and OPTIM_REPORT_PWRUSR services were up and available on node 5 at 6:51am (1 minute outage).
All Nodes restarted by 07:05am Dec 7th
All Loaders restarted at 07:07am Dec 7th
All Summaries restarted at 09:54am Dec 7th

Root Cause Analysis Doc (6:7)

3 Initial RCA Statement

With the current settings on the system Node 5 has the highest load in the system. When the resource manager was disabled there were periods when resource manager was off where the system was working at 95% CPU load. User queries were executing on the system with hints having a high degree of parallel requested. It is currently believed that high parallel query at the same time as a high sustained CPU load caused the initial problem leading to the node eviction.

The initial findings point to the fact that by turning off the RM and keeping the parallel max_server fairly high at 300 per instance resulted in Oracle hanging after a period of time as there was an increase in parallel sessions which in turn consumed all available cpu and resulted in timeouts for important tasks like inter node communication or slow response to other nodes.

When the DBAs tried to bring back node 5,the Oracle Database tried to rebalance that again, and caused all the process that access cluster information like GV$ views to hang on "reliable message". This is believed to be a bug previously experienced on the system for Node 2 on 07-Mar-2010 (Bug 9453004: INVESTIGATE DB INSTANCE 2 AND NODE B REBOOT). The inability to rebalance the Database meant the system could not come up cleanly as a 4 node RAC environment which is why a Database Bounce was requested.

Root Cause Analysis Doc (7:7)

4 Conclusion / Next Steps

Await response from Oracle RCA (bug 103776604) following their analysis of trace files/dumps taken during outage.

Recommend that Oracle Bug 9453004 raised March 7 be fixed by Oracle and applied when available.

User parallel query hints should be reduced and if at all possible eliminated. The loads on some nodes are high and may mean that we are under resourced and require extra hardware. Even with Resource Manager in place this may result in sub-optimal query times for users.

Re-assess if PSU need to be applied more regularly to minimize exposure to Oracle Bugs.

It is recommended that if two DBAs are working on the system at the same time, that they are in full communications with each other.

Explore the following potential related bugs:

Bug 6148054 - RAC hang waiting for "reliable message" [ID 6148054.8]

Bug 6773260: ORA-00600 [KOKEGPINLOB1] ON SELECT FROM GV$PX_SESSION, GV$SESSION AND GV$SQLAREA

Let's Examine Some Real-World Cases

- Case 1: The Puppet Master
- Case 2: Jobs and Human Nature
- Case 3: More Jobs and Human Nature
- Case 4: Port Exhaustion
- Case 5: Storage Storage Everywhere
- Case 6: UCS (Unimpressive Common Servers)
- Case 7: Cisco 5010 to 7010 Migration
- Case 8: RAC is RAC: Server Manager is not Optional

Case 1: The Puppet Master

Fingerprint At The Scene of the Crime

Two physically separate two-node RAC clusters. They do not share servers. They do not share storage. They do not share network any component of the cache fusion interconnect ... and yet ... in 7 minutes 7 seconds ... they both go down.

Is it a database problem?

DC20PCE11

```
Thu Aug 08 16:52:30 2013 Archived Log entry 215974 added for thread 1 sequence 216019 ID 0x2d7ba8f dest 1:
Thu Aug 08 16:57:27 2013 Time drift detected. Please check VKTM trace file for more details.
Thu Aug 08 16:57:43 2013 ERROR: unrecoverable error ORA-29701 raised in ASM I/O path; terminating process 12257
```

DS20SCE11

```
Thu Aug 08 16:57:17 2013 Completed checkpoint up to RBA [0xae7f.2.10], SCN: 780145612
Thu Aug 08 17:04:34 2013 Time drift detected. Please check VKTM trace file for more details.
Thu Aug 08 17:04:46 2013 ERROR: unrecoverable error ORA-29701 raised in ASM I/O path; terminating process 2445
```

Production CRSD Log: 4 Seconds Earlier

ORAP1N1

```
2013-08-08 16:57:31.162: [    AGFW][1164335424] {0:12:9} Agfw Proxy Server received the message:
RESOURCE_STATUS[Proxy] ID 20481:147794
2013-08-08 16:57:31.162: [    AGFW][1164335424] {0:12:9} Received state change for ora.LISTENER_SCAN2.lsnr 1 1
[old state = ONLINE, new state = OFFLINE]
```

ORAP1N2

```
2013-08-08 17:09:09.393: [UiServer][1175996736] {2:7473:48658} Done for ctx=0x2aaaac2532b0
2013-08-08 17:09:39.156: [GIPCHDEM][1115060544] gipchaDaemonProcessHAInvalidate: completed ha name invalidate
for node 0x2aaaac25bb60 { host 'orap1n1', haName '9f34-b767-de19-a294', srcLuid 04a03a5c-f4851208, dstLuid
e3aa430e-82601c00 numInf 2, contigSeq 62781, lastAck 56961, lastValidAck 62780, sendSeq [56961 : 56961],
createTime 72155204, flags 0x28 }
```

P1N1 to P1N2 issue delta: 12 minutes 8 seconds

Staging CRSD Log: 4 Hours 11 Minutes Earlier

ORAS1N1

```
2013-08-08 13:04:45.315: [    AGFW][1159891264] {0:4:7} Agfw Proxy Server received the message:
RESOURCE_STATUS[Proxy] ID 20481:508508
2013-08-08 13:04:45.315: [    AGFW][1159891264] {0:4:7} Received state change for ora.asm oras1n1 1 [old state
= ONLINE, new state = UNKNOWN]
```

ORAS1N2

```
2013-08-08 13:12:07.199: [ CRSMAIN][96481872] Sync-up with OCR
2013-08-08 13:12:07.199: [ CRSMAIN][96481872] Connecting to the CSS Daemon
2013-08-08 13:12:07.202: [  CRSRTI][96481872] CSS is not ready. Received status 3
2013-08-08 13:12:07.202: [ CRSMAIN][96481872] Created alert : (:CRSD00109:) :  Could not init the CSS context,
error: 3
2013-08-08 13:12:07.202: [    CRSD][96481872][PANIC] CRSD exiting: Could not init the CSS context, error: 3
```

S1N1 to S1N2 issue delta: 7 minutes 22 seconds

Staging CSSD Log From S1N1

```
2013-08-08 17:07:42.784: [    CSSD][1113848128]clssnmSendingThread: sending status msg to all nodes
2013-08-08 17:07:42.784: [    CSSD][1113848128]clssnmSendingThread: sent 4 status msgs to all nodes
2013-08-08 17:07:46.595: [    CSSD][1101232448]clssgmUnregisterShared: Same group share client 1 (0x2aaab0242700), grp DG_LOCAL_DG01, member 0
2013-08-08 17:07:46.596: [    CSSD][1101232448]clssgmTermShare: (0x2aaab00468e0) local grock DG_LOCAL_DG01 member 0 type 1
2013-08-08 17:07:46.596: [    CSSD][1101232448]clssgmUnreferenceMember: local grock DG_LOCAL_DG01 member 0 refcount is 19
2013-08-08 17:07:46.596: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8ed7f71
2013-08-08 17:07:46.601: [    CSSD][1101232448]clssgmUnregisterShared: Same group share client 1 (0xf0701e0), grp DG_LOCAL_DG01, member 0
2013-08-08 17:07:46.601: [    CSSD][1101232448]clssgmTermShare: (0xf0df060) local grock DG_LOCAL_DG01 member 0 type 1
2013-08-08 17:07:46.601: [    CSSD][1101232448]clssgmUnreferenceMember: local grock DG_LOCAL_DG01 member 0 refcount is 18
2013-08-08 17:07:46.602: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8f85cc6
2013-08-08 17:07:46.605: [    CSSD][1101232448]clssgmUnregisterShared: Same group share client 2 (0x2aaab0201a20), grp DG_LOCAL_DG02, member 0
2013-08-08 17:07:46.605: [    CSSD][1101232448]clssgmTermShare: (0x2aaab0462f30) local grock DG_LOCAL_DG02 member 0 type 1
2013-08-08 17:07:46.605: [    CSSD][1101232448]clssgmUnreferenceMember: local grock DG_LOCAL_DG02 member 0 refcount is 12
2013-08-08 17:07:46.605: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8e510f7
2013-08-08 17:07:46.608: [    CSSD][1101232448]clssgmUnregisterShared: Same group share client 2 (0x2aaab0429d00), grp DG_LOCAL_DG02, member 0
2013-08-08 17:07:46.609: [    CSSD][1101232448]clssgmTermShare: (0x2aaab0261ec0) local grock DG_LOCAL_DG02 member 0 type 1
2013-08-08 17:07:46.609: [    CSSD][1101232448]clssgmUnreferenceMember: local grock DG_LOCAL_DG02 member 0 refcount is 11
2013-08-08 17:07:46.609: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8f33786
2013-08-08 17:07:46.611: [    CSSD][1101232448]clssgmDeadProc: proc 0x2aaab01a1ff0
2013-08-08 17:07:46.611: [    CSSD][1101232448]clssgmDestroyProc: cleaning up proc(0x2aaab01a1ff0) con(0x8ed7f39) skgpid 19273 ospid 19273 with 0 clients, refcount 0
2013-08-08 17:07:46.611: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8ed7f39
2013-08-08 17:07:46.613: [    CSSD][1101232448]clssgmUnregisterShared: Same group share client 1 (0x2aaab05885e0), grp DG_LOCAL_DG01, member 0
2013-08-08 17:07:46.613: [    CSSD][1101232448]clssgmTermShare: (0x2aaab01f52f0) local grock DG_LOCAL_DG01 member 0 type 1
2013-08-08 17:07:46.613: [    CSSD][1101232448]clssgmUnreferenceMember: local grock DG_LOCAL_DG01 member 0 refcount is 17
2013-08-08 17:07:46.613: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8e510d8
2013-08-08 17:07:46.616: [    CSSD][1101232448]clssgmUnregisterShared: Same group share client 1 (0x2aaab04eb8b0), grp DG_LOCAL_DG01, member 0
2013-08-08 17:07:46.616: [    CSSD][1101232448]clssgmTermShare: (0x2aaab0430400) local grock DG_LOCAL_DG01 member 0 type 1
2013-08-08 17:07:46.616: [    CSSD][1101232448]clssgmUnreferenceMember: local grock DG_LOCAL_DG01 member 0 refcount is 16
2013-08-08 17:07:46.616: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8f33736
2013-08-08 17:07:46.618: [    CSSD][1101232448]clssgmDeadProc: proc 0xf13cba0
2013-08-08 17:07:46.618: [    CSSD][1101232448]clssgmDestroyProc: cleaning up proc(0xf13cba0) con(0x8f85c8e) skgpid 18514 ospid 18514 with 0 clients, refcount 0
2013-08-08 17:07:46.618: [    CSSD][1101232448]clssgmDestroyProc: cleaning up proc(0xf13cba0) con(0x8f85c8e) skgpid 18514 ospid 18514 with 0 clients, refcount 0
2013-08-08 17:07:46.618: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8f85c8e
2013-08-08 17:07:46.622: [    CSSD][1101232448]clssgmDeadProc: proc 0x2aaab030da10
2013-08-08 17:07:46.622: [    CSSD][1101232448]clssgmDestroyProc: cleaning up proc(0x2aaab030da10) con(0x8e510a0) skgpid 4727 ospid 4727 with 0 clients, refcount 0
2013-08-08 17:07:46.622: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8e510a0
2013-08-08 17:07:46.625: [    CSSD][1101232448]clssgmDeadProc: proc 0x2aaab04e9010
2013-08-08 17:07:46.625: [    CSSD][1101232448]clssgmDestroyProc: cleaning up proc(0x2aaab04e9010) con(0x8f336fe) skgpid 25768 ospid 25768 with 0 clients, refcount 0
2013-08-08 17:07:46.625: [    CSSD][1101232448]clssgmDiscEndpcl: gipcDestroy 0x8f336fe
2013-08-08 17:07:47.795: [    CSSD][1113848128]clssnmSendingThread: sending status msg to all nodes
```

O/S Log: Four Days Before Incident

```
Aug  4 04:09:16 orap1n1 Updating DNS configuration for: orap1n1.lux20.morgan.priv
Aug  4 04:09:16 orap1n1 Initial DNS Server: 10.2.198.34
Aug  4 04:09:16 orap1n1 Connecting to DNS server 10.2.198.34
Aug  4 04:09:16 orap1n1 Connected to DNS server 10.2.198.34
Aug  4 04:09:16 orap1n1 Updating both HOST and PTR record  for: orap1n1.lux2.morgan.priv
Aug  4 04:09:16 orap1n1 Deleting old reverse lookup records for orap1n1.lux2.morgan.priv on 10.2.198.34.
Aug  4 04:09:17 orap1n1 Adding GSS support to DNS server 10.2.198.34
Aug  4 04:09:17 orap1n1 Added GSS support to DNS server 10.2.198.34
Aug  4 04:09:17 orap1n1 Failed to delete reverse lookup record 11.78.2.10.in-addr.arpa. Reason Refused (5).
Aug  4 04:09:17 orap1n1 Deleting reverse lookup records for our current new IP Address(s) on
ad010.lux20.morgan.priv.
Aug  4 04:09:18 orap1n1 No reverse lookup records found for 11.0.168.192.in-addr.arpa on ad010.ams20.morgan.priv.
Aug  4 04:09:18 orap1n1  No reverse lookup records found for 21.34.254.169.in-addr.arpa on ad010.lux20.morgan.priv.
Aug  4 04:09:19 orap1n1 No reverse lookup records found for 12.0.168.192.in-addr.arpa on ad010.lux20.morgan.priv.
Aug  4 04:09:20 orap1n1 No reverse lookup records found for 181.139.254.169.in-addr.arpa on ad010.lux20.morgan.priv.
Aug  4 04:09:20 orap1n1 Failed to delete reverse lookup record 11.78.2.10.in-addr.arpa. Reason Refused (5).
Aug  4 04:09:21 orap1n1 Failed to delete reverse lookup record 10.78.2.10.in-addr.arpa. Reason Refused (5).
Aug  4 04:09:22 orap1n1 Failed to delete reverse lookup record 102.78.2.10.in-addr.arpa. Reason Refused (5).
Aug  4 04:09:22 orap1n1 Failed to delete reverse lookup record 100.78.2.10.in-addr.arpa. Reason Refused (5).
Aug  4 04:09:23 orap1n1 Failed to delete reverse lookup record 14.2.2.10.in-addr.arpa. Reason Refused (5).
Aug  4 04:09:23 orap1n1 Deleting host records for orap1n1.lux20.morgan.priv on ad010.lux20.morgan.priv.
Aug  4 04:09:23 orap1n1 Failed to delete host record for orap1n1.lux20.morgan.priv. Reason Refused (5).
```

7,824 lines of changes in /var/log/messages on one server
This happened 152 times on ORAP1N1, in DC20, in 6 days

OS Log: Two Days After Incident

```
Aug 10 12:03:23 orap1n1 Updating DNS configuration for: orap1n1.lux20.morgan.priv
Aug 10 12:03:23 orap1n1 Initial DNS Server: 10.2.198.33
Aug 10 12:03:23 orap1n1 Connecting to DNS server 10.2.198.33
Aug 10 12:03:23 orap1n1 Connected to DNS server 10.2.198.33
Aug 10 12:03:24 orap1n1 Updating both HOST and PTR record for: orap1n1.lux20.morgan.priv
Aug 10 12:03:24 orap1n1 Deleting old reverse lookup records for orap1n1.lux20.morgan.priv on 10.2.198.33.
Aug 10 12:03:24 orap1n1 Adding GSS support to DNS server 10.2.198.33
Aug 10 12:03:24 orap1n1 Added GSS support to DNS server 10.2.198.33
Aug 10 12:03:25 orap1n1 Failed to delete reverse lookup record 11.78.2.10.in-addr.arpa. Reason Refused (5).
Aug 10 12:03:25 orap1n1 Deleting reverse lookup records for our current new IP Address(s) on ad009.lux20.morgan.priv.
Aug 10 12:03:25 orap1n1 No reverse lookup records found for 11.0.168.192.in-addr.arpa on ad009.lux20.morgan.priv.
Aug 10 12:03:26 orap1n1 No reverse lookup records found for 21.34.254.169.in-addr.arpa on ad009.lux20.morgan.priv.
Aug 10 12:03:27 orap1n1 No reverse lookup records found for 12.0.168.192.in-addr.arpa on ad009.lux20.morgan.priv.
Aug 10 12:03:27 orap1n1 No reverse lookup records found for 181.139.254.169.in-addr.arpa on ad009.lux20.morgan.priv.
Aug 10 12:03:28 orap1n1 Failed to delete reverse lookup record 11.78.2.10.in-addr.arpa. Reason Refused (5).
Aug 10 12:03:28 orap1n1 Failed to delete reverse lookup record 10.78.2.10.in-addr.arpa. Reason Refused (5).
Aug 10 12:03:29 orap1n1 Failed to delete reverse lookup record 101.78.2.10.in-addr.arpa. Reason Refused (5).
Aug 10 12:03:30 orap1n1 Failed to delete reverse lookup record 14.2.2.10.in-addr.arpa. Reason Refused (5).
Aug 10 12:03:30 orap1n1 Deleting host records for orap1n1.lux20.morgan.priv on ad009.lux20.morgan.priv.
Aug 10 12:03:30 orap1n1 Failed to delete host record for orap1n1.lux20.morgan.priv. Reason Refused (5).
Aug 10 12:03:30 orap1n1 Updating host records for orap1n1.lux20.morgan.priv on ad009.lux20.morgan.priv.
Aug 10 12:03:31 orap1n1 Failed to update host records orap1n1.lux20.morgan.priv: Reason Refused (5).
```

Log File Research

To: <system and storage admins>
Cc: <management>
Subject: Need Hardware Information

Here is the signature of the ASM failure in DC20 over the last two days. Two different databases on different blades:

```
*** 2013-08-09 11:49:20.023
NOTE: ASMB terminating
ORA-15064: communication failure with ASM instance
ORA-03113: end-of-file on communication channel
Process ID:
Session ID: 82 Serial number: 9
error 15064 detected in background process
ORA-15064: communication failure with ASM instance
ORA-03113: end-of-file on communication channel
Process ID:
Session ID: 82 Serial number: 9
kjzduptcctx: Notifying DIAG for crash event
----- Abridged Call Stack Trace -----
ksedsts()+461<-kjzdssdmp()+267<-kjzduptcctx()+232<-kjzdicrshnfy()+53<-ksuitm()+1325<-
ksbrdp()+3344<-opirip()+623<-opidrv()+603<-sou2o()+103<-opimai_real()+266<-ssthrdmain()+252<-
main()+201<-__libc_start_main()+244<-_start()+36
----- End of Abridged Call Stack Trace -----

*** 2013-08-09 11:49:20.134
ASMB (ospid: 15341): terminating the instance due to error 15064
ksuitm: waiting up to [5] seconds before killing DIAG(15317)
```

Can you help me please with the following:

1. Are all database blades in the same or different chassis?
2. What is the storage solution? VSP, NetApp? What diagnostics can we pull?
3. What network infrastructure between the blades and the storage array? What diagnostics can we pull?

O/S Log: Ruby on RAC?

```
Aug  8 13:04:22 orap1n1 ERROR:  While executing gem ... (Gem::RemoteFetcher::FetchError)
Aug  8 13:04:22 orap1n1 Errno::ETIMEDOUT: Connection timed out - connect(2) (http://rubygems.org/latest_specs.4.8.gz)
Aug  8 13:04:22 orap1n1 INFO:  `gem install -y` is now default and will be removed
Aug  8 13:04:22 orap1n1 INFO:  use --ignore-dependencies to install only the gems you list


Aug  8 15:42:41 orap1n1 ERROR:  While executing gem ... (Gem::RemoteFetcher::FetchError)
Aug  8 15:42:41 orap1n1 Errno::ETIMEDOUT: Connection timed out - connect(2) (http://rubygems.org/latest_specs.4.8.gz)
Aug  8 15:42:41 orap1n1 INFO:  `gem install -y` is now default and will be removed
Aug  8 15:42:41 orap1n1 INFO:  use --ignore-dependencies to install only the gems you list
```

This happened twice just before the outage
the first time 3 hours 53 seconds before the outage

The second time 1 hour 15 minutes before the outage

O/S Log: NTP Time Synchronization

```
Aug  8 12:56:04 orap1n1 ntpd[1339]: ntpd exiting on signal 15
Aug  8 12:57:27 orap1n1 ntpdate[12406]: step time server 10.2.255.254 offset 82.262906 sec
Aug  8 12:57:27 orap1n1 ntpd[12408]: ntpd 4.2.2p1@1.1570-o Fri Jul 22 18:07:53 UTC 2011 (1)
Aug  8 12:57:27 orap1n1 ntpd[12409]: precision = 1.000 usec
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface wildcard, 0.0.0.0#123 Disabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface wildcard, ::#123 Disabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface bond2, fe80::217:a4ff:fe77:fc18#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface lo, ::1#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface bond0, fe80::217:a4ff:fe77:fc10#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface eth2, fe80::217:a4ff:fe77:fc14#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface eth3, fe80::217:a4ff:fe77:fc16#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface lo, 127.0.0.1#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface eth2, 192.168.0.11#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface eth2:1, 169.254.34.21#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface eth3, 192.168.0.12#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface eth3:1, 169.254.139.181#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface bond0, 10.2.78.11#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface bond0:1, 10.2.78.10#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface bond0:3, 10.2.78.102#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface bond0:4, 10.2.78.100#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: Listening on interface bond2, 10.2.2.14#123 Enabled
Aug  8 12:57:27 orap1n1 ntpd[12409]: kernel time sync status 0040
Aug  8 12:57:27 orap1n1 ntpd[12409]: frequency initialized 0.000 PPM from /var/lib/ntp/drift
```

The Puppet Master Conclusions

- Two physically distinct RAC clusters were brought down by the use of a totally inappropriate tool by people that did not understand the implications of what they were doing
- The DBA team was never informed that the activity was going to take place
- The DBA team had no access to the change logs
- The databases were innocent bystanders and there was nothing the DBA team could have done to prevent the outage
- IT management was at fault for the outage by creating a situation with
  - Inadequate communications
  - Inadequate training for non-DBAs

Case 2: Jobs and Human Nature

Repeating Issue: User Configured Loads

RAC Server Node 1

MMDD	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0804	0	0	0	0	0	0	0	0	0	0	0	0	5	32	18	65	91	13	12	20	84	9	14	9
0805	137	112	26	27	141	17	21	9	85	13	21	17	96	23	23	24	91	13	11	21	86	11	14	9
0806	151	111	21	24	96	41	50	14	84	22	20	22	91	18	17	18	92	24	10	11	83	9	14	20
0807	139	100	32	30	99	43	49	19	105	17	31	14	76	23	27	25	111	20	15	18	86	13	13	10
0808	145	99	29	30	109	52	48	11	102	25	47	24	101	23	20	23	117	31	30	16	91	12	11	9
0809	123	83	65	37	93	17	25	10	102	23	44	25	111	37	24	29	98	19	29	16	92	16	15	9
0810	169	120	52	32	125	58	38	9	109	17	26	14	104	13	17	15	93	13	16	11	61	10	10	9
0811	107	82	51	34	85	17	22	10	73	10	12	11	92	32	13	69	65	11	11	10	60	9	12	9
0812	149	121	26	15	70	16	24	11	95	34	15	18	34	67	21	21	87	11	13	9	77	9	14	9
0813	115	76	55	56	27	9	9	9	11	9	9	0	0	0	0	0	0	0	0	0	0	0	0	0

60 corresponds to one change per minute ... the ideal range is 4 to 12
Addressed by resizing redo logs from 400MB to 4GB
And rescheduling many of the jobs

Jobs And Human Nature Conclusions

- DBAs must be able to control when and where jobs are scheduled based on run time and resource requirements
- Applications should NEVER be allowed to schedule multiple jobs to start at the exact same point-in-time
- Developers need to have partial ownership the problem
- None of this is likely to ever happen so DBAs need to play defense and monitor job scheduling by creating their own alerts
- IT management was at fault for the performance hits due to
  - Inadequate communications
  - Inadequate training for non-DBAs

Case 3: More Jobs and Human Nature

Unobserved Job Failure

```
SQL> SELECT owner, job_name, job_type, trunc(start_date) SDATE, trunc(next_run_date) nxtrun, failure_count
  2  FROM dba_scheduler_jobs
  3* WHERE failure_count <> 0;

OWNER           JOB_NAME                        STATE      SDATE                NXTRUN               FAILURE_COUNT
--------------- ------------------------------- ---------- -------------------- -------------------- -------------
SYS             SM$CLEAN_AUTO_SPLIT_MERGE       SCHEDULED  14-MAR-2011 00:00:00 14-AUG-2013 00:00:00            17
SYS             RSE$CLEAN_RECOVERABLE_SCRIPT    SCHEDULED  14-MAR-2011 00:00:00 14-AUG-2013 00:00:00            20
SYS             DRA_REEVALUATE_OPEN_FAILURES    SCHEDULED                                                        10
ORACLE_OCM      MGMT_CONFIG_JOB                 SCHEDULED                                                         4
EXFSYS          RLM$SCHDNEGACTION               SCHEDULED  13-AUG-2013 00:00:00 13-AUG-2013 00:00:00             3
EXFSYS          RLM$EVTCLEANUP                  SCHEDULED  27-APR-2011 00:00:00 13-AUG-2013 00:00:00             2
RDBA5           LONG_RUN_SESS_JOB               SCHEDULED  12-AUG-2013 00:00:00 13-AUG-2013 00:00:00             1
EISAI_PROD_TMS  POPULATE_MORGAN_CATALOG         DISABLED   01-JUN-2009 00:00:00 08-AUG-2013 00:00:00          2559
```

Another Unobserved Job Failure

```
SQL> SELECT owner, job_name, job_type, state, trunc(start_date) SDATE, trunc(next_run_date) NXTRUN, failure_count
  2  FROM dba_scheduler_jobs
  3  WHERE failure_count > 0
  4* ORDER BY 6;

OWNER           JOB_NAME                       STATE      SDATE                NXTRUN               FAILURE_COUNT
--------------- ------------------------------ ---------- -------------------- -------------------- -------------
SYS             PVX_STUDENT                    SCHEDULED  29-MAR-2013          09-AUG-2013                   122
```

Called out in Jira CO-9060 for the following exception:

r-succe-ds:aukoras1n4 Logscan matched patterns in /app/oracle/base/diag/rdbms/auksce54/AUKSCE541/trace/alert_AUKSCE541.log RDBA WARN + Errors in file /app/oracle/base/diag/rdbms/auksce54/AUKSCE541/trace/AUKSCE541_j000_12172.trc: + ORA-12012: error on auto execute of job "SYS"."PVX_STUDENT_REFRESH" W ORA-06550: line 1, column 797: + PLS-00103: Encountered the symbol "PVX_STUDENT" when expecting one of the following: + +) , * & = - + < / > at in is mod remainder not rem => + <> or != or ~= >= <= <> and or like like2 + like4 likec as between from using || multiset member ---------- alert.pl v5.3.120207 mon_hub:auktusc01 (auktusc01) run_time:2013-Aug-08 08:01:43 client:R-SUCCE-DS server:auktusc01 entity:aukoras1n4 entity_type:OPSYS processed by ftp_mail_proc.pl on delphi at 8-Aug-2013 07:03

Jobs And Human Nature Conclusions

- Sorry DBAs you don't get off blameless on this one
- As a DBA you MUST know what is happening on your system and you need to monitor more than just your email and the alert log
  - /var/log/messages
  - alert log
  - clusterware logs
  - ASM log
  - listener log
  - backup logs
  - job scheduling logs
  - AUD$ and FGA$
  - SQL*Loader and External table loading logs
  - DataPump logs
  - And no you can not rely on OEM Cloud Control for most of this

Case4: Port Exhaustion

Hint: It is not caused by drinking too much port

In The Beginning (1:4)

- Customer Reports are stuck in the queue

Hi Ops

Report Jobs are getting stuck in Waiting in Queue. Also, having performance issues with Admin side

Thanks,
J

Step to Recreate

1. Log into Website
2. Navigate to Reports
3. Search for Account Data
4. Run the report for morgand
5. Notice that the report is stuck in Waiting in Queue

In The Beginning (2:4)

- The website generated an HTTP403 error

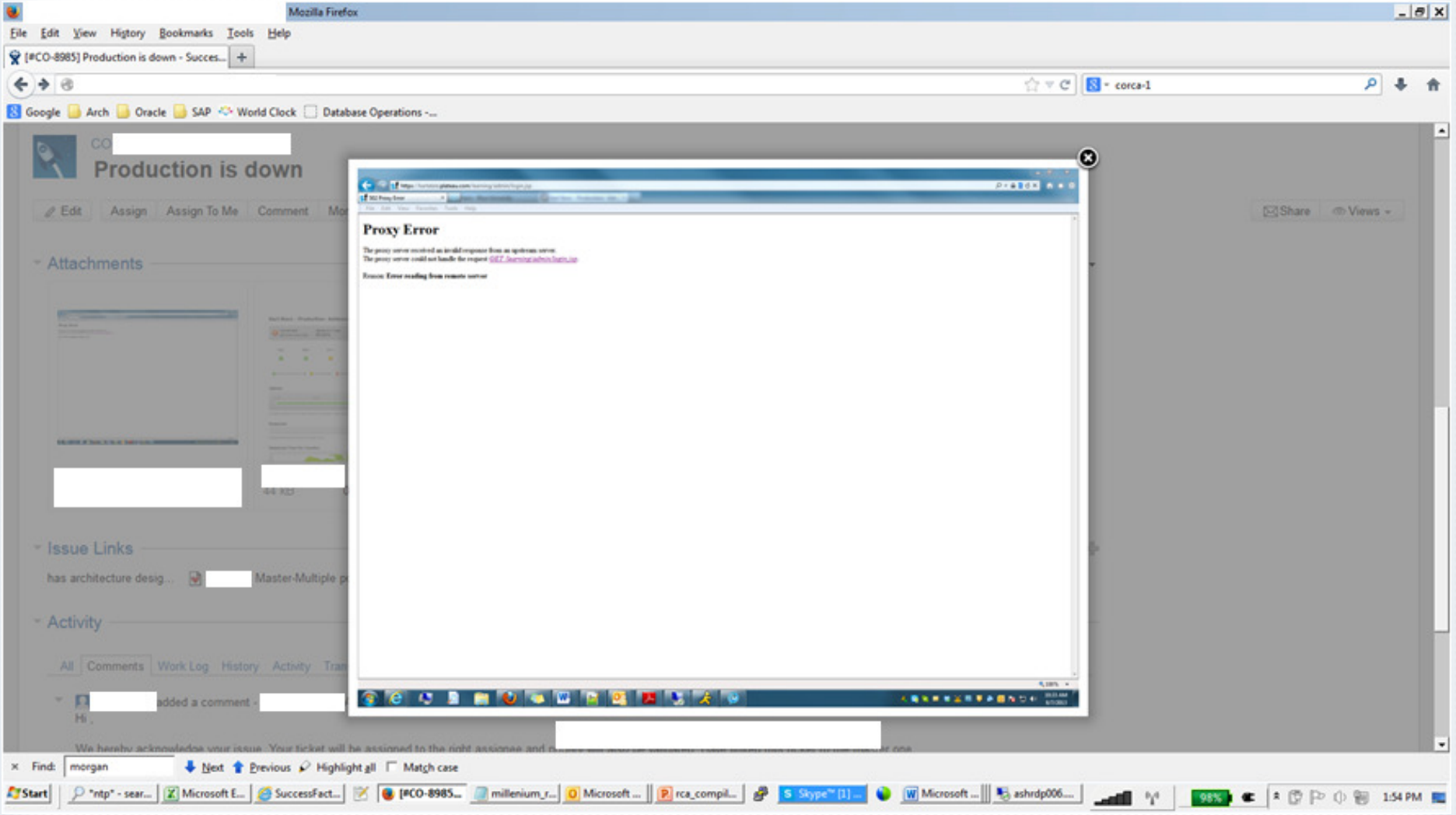
As a partner we got communication that the previously assigned sandboxes will be brought down.

Instead -as a partner- we got a them demo environment assigned (Tenant ID: PARTNER0001 which we have integrated with a customer database instance (xxxdemo ace4morgan).

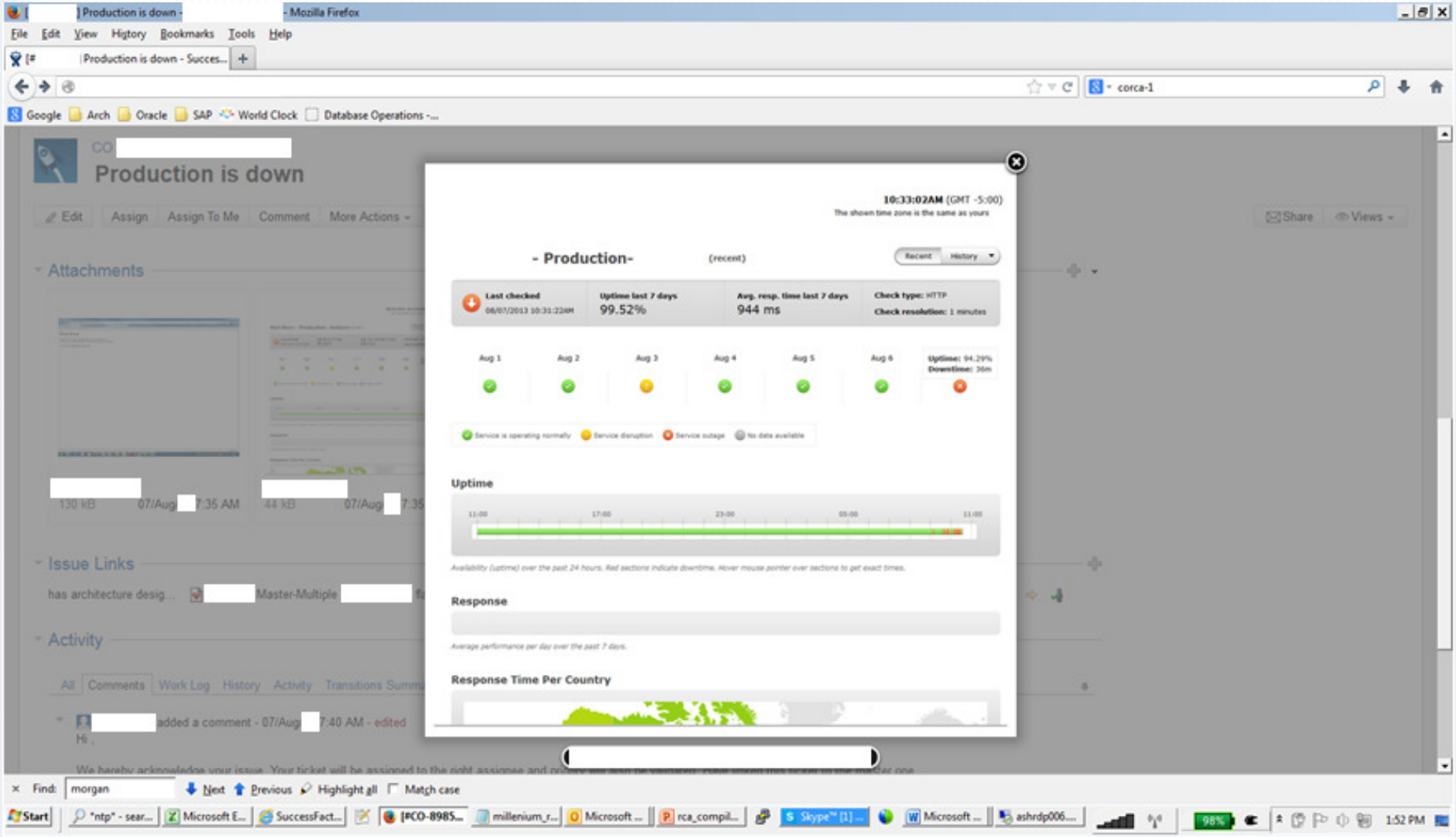
Everything was working fine (including integration).
Today I tried to access the instance via the partner and via the direct url (https://partner0001.demo.xxx.com/admin/nativelogin.jsp) but in both case an error is displayed on the screen (see attachment).

We need this be fixed as soon as possible!
(major customer demo session on Friday!)

In The Beginning (3:4)



In The Beginning (4:4)

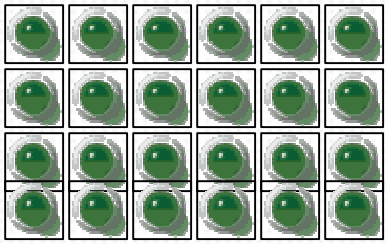


How Does An Application Server Connect to RAC?

- Do you connect to the SCAN IP by name or number?
- If a name ... a DNS server resolves the name to an IP
- To avoid single points of failure you should have two or more DNS servers with a load balancer, or two, in front of them
- The SCAN IP points to a VIP which may again need to be resolved from a name to a physical IP address
- The VIP may again point to a name which must be resolved to a physical IP address
- Most servers cache DNS entries to improve speed
  - Do you know if yours do?

Triaging a Connection Problem

- Try to connect to the cluster?
  - From where?
  - With what tool?
  - To the SCAN, VIP, or physical IP?
- Ping the IP addresses
- Run Trace Route on the IP addresses
- Read the listener log
- Read the database alert log
- Call the network admins who will tell you



everything looks good ...
the network is just Ok ...
the network is always Ok
the network will always be Ok

RESOLVE.CONF (1:3)

NAME

resolv.conf- resolver configuration file

SYNOPSIS

`/etc/resolv.conf`

DESCRIPTION

The `resolver` is a set of routines that provide access to the Internet Domain Name System. See **resolver(3RESOLV)**. `resolv.conf` is a configuration file that contains the information that is read by the `resolver` routines the first time they are invoked by a process. The file is designed to be human readable and contains a list of keywords with values that provide various types of `resolver` information.

The `resolv.conf` file contains the following configuration directives:

`nameserver`

Specifies the Internet address in dot-notation format of a name server that the resolver is to query. Up to *MAXNS* name servers may be listed, one per keyword. See `<resolv.h>`. If there are multiple servers, the resolver library queries them in the order listed. If no name server entries are present, the resolver library queries the name server on the local machine. The resolver library follows the algorithm to try a name server until the query times out. It then tries the the name servers that follow, until each query times out. It repeats all the name servers until a maximum number of retries are made.

`domain`

Specifies the local domain name. Most queries for names within this domain can use short names relative to the local domain. If no domain entry is present, the domain is determined from **sysinfo(2)** or from gethostname(3C). (Everything after the first `.' is presumed to be the domain name.) If the host name does not contain a domain part, the root domain is assumed. You can use the `LOCALDOMAIN` environment variable to override the domain name.

RESOLVE.CONF (2:3)

`search`

The search list for host name lookup. The search list is normally determined from the local domain name. By default, it contains only the local domain name. You can change the default behavior by listing the desired domain search path following the search keyword, with spaces or tabs separating the names. Most `resolver` queries will be attempted using each component of the search path in turn until a match is found. This process may be slow and will generate a lot of network traffic if the servers for the listed domains are not local. Queries will time out if no server is available for one of the domains.

The search list is currently limited to six domains and a total of 256 characters.

`sortlist` ***addresslist***

Allows addresses returned by the libresolv-internal `gethostbyname()` to be sorted. A `sortlist` is specified by IP address netmask pairs. The netmask is optional and defaults to the natural netmask of the net. The IP address and optional network pairs are separated by slashes. Up to 10 pairs may be specified. For example:

```
sortlist 130.155.160.0/255.255.240.0 130.155.0.0
```

RESOLVE.CONF (3:3)

`options`

Allows certain internal resolver variables to be modified. The syntax is

```
options option ...
```

where option is one of the following:

`debug`

Sets `RES_DEBUG` in the `_res.options` field.

`ndots:` ***n***

Sets a threshold floor for the number of dots which must appear in a name given to `res_query()` before an initial absolute (as-is) query is performed. See **resolver(3RESOLV)**. The default value for *n* is 1, which means that if there are any dots in a name, the name is tried first as an absolute name before any search list elements are appended to it.

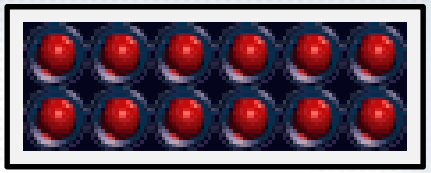
`timeout:` ***n***
`retrans:` ***n***

Sets the amount of time the resolver will wait for a response from a remote name server before retrying the query by means of a different name server. Measured in seconds, the default is `RES_TIMEOUT`. See `<resolv.h>`. The `timeout` and `retrans` values are the starting point for an exponential back off procedure where the `timeout` is doubled for every retransmit attempt.

`attempts:` ***n***
`retry:` ***n***

Sets the number of times the resolver will send a query to its name servers before giving up and returning an error to the calling application. The default is `RES_DFLRETRY`. See `<resolv.h>`.

Resolution: The DNS Admin POV (1:3)



On August 7th, we experienced a 2 hour outage that impacted more than 150 customers. In researching this outage it was noticed that DNS caching had been disabled on the Oracle Database Servers. Also, in going through the logs on the F5 Local Traffic Manager (LTM), is was noticed that there were 39,696 port exhaustion errors on port 53 going to the three DNS servers, starting at approximately 4am and ending slightly after 3pm. There were also an additional 625,665 port exhaustion error messages that were dropped in the logs, bringing the total to 665,361 port exhaustion error messages.

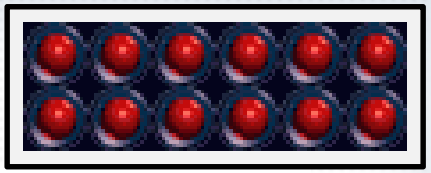
Further research discovered that there was a misconfiguration in the resolv.conf file on the servers in the data center. The resolv.conf file on these servers looked like this:

```
search morgan.priv
nameserver 10.24.244.200 (VIP pointing to servers listed below)
nameserver 10.24.244.21 (Bind server 01)
nameserver 10.24.244.25 (Bind server 02)
nameserver 10.24.244.29 (Bind server 03)
```

This results in the first DNS query going to the VIP for hostname and reverse IP resolution, and then to the three DNS servers. However, the 3 DNS servers which were supposed to be the alternative option to the VIP are also pointing to the same VIP. This basically sets up an infinite loop until the DNS queries time out.

The recommended resolution was to remove the VIP and have the servers query the DNS servers directly.

Resolution: The DNS Admin POV (2:3)



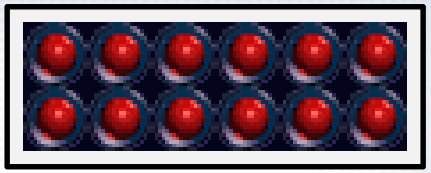
These graphs give an overview of what was happening throughout August 7th on the servers. I noticed that there is a sudden drop in connections right around 10:40am; and returning at around 10:45 am.

If you look at the files I've sent out previously, there is actually less evidence of port exhaustion between 10:22 and 10:42; with increasing levels of port exhaustion as connections and activity increases after about 12:07pm.

Additionally, I went back over the last few days and looked for port exhaustion for the DNS servers on port 53 and found the following:

```
Jul 29 -     16 port exhaustion errors
Jul 30 -      7 port exhaustion errors
Jul 31 -      8 port exhaustion errors
Aug 1  -      6 port exhaustion errors
Aug 2  - 38,711 port exhaustion errors
Aug 3  - 26,023 port exhaustion errors
Aug 4  - 22,614 port exhaustion errors
Aug 5  -     20 port exhaustion errors
Aug 6  - 11,282 port exhaustion errors
```

Resolution: The DNS Admin POV (3:3)



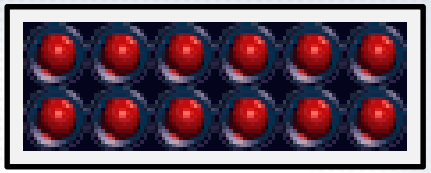
Additionally, I did some calculations on the additional port exhaustion log messages that were dropped – these were the throttling error that I mentioned previously.

On the 7th of August there were an additional 625,665 port exhaustion error messages that were dropped. On August 3rd, there were an additional 99,199 port exhaustion error messages that were dropped.

And on August 2nd, there were an additional 204,315 port exhaustion error messages that were dropped.

These numbers are in addition to the numbers of port exhaustion errors previously reported.

Resolution: The System Admin POV



Every unix box at the LAX data center has this resolv.conf file:

```
search morgan.priv
nameserver 10.24.244.200 (VIP pointing to both AD01 and AD02 windows servers)
nameserver 10.24.244.21  (Bind server 01)
nameserver 10.24.244.25  (Bind server 02)
nameserver 10.24.244.29  (Bind server 03)
```

The idea behind this design is to firstly query the VIP (for hostname resolution) and then, the 3 bind servers which are slave DNS servers of the AD DNS servers described above.

Now, I've found that the BIND servers (unix) which are supposed to be the alternative option to the VIP, have the same /etc/resolv.conf file and therefore are also pointing to the VIP on the first place. As you can imagine this basically ends up in an infinite loop until the load balancer get finally some relief or the DNS queries timeout.

Refer to the attachment "Morgan current arch" to see the workflow.

The fix should be easy and basically would consist of removing the VIP from the /etc/resolv.conf from the Bind servers and have them pointing to each AD server (bind01 -> AD01, bind02 -> AD02, etc).

The ultimate solution would be to remove the VIP from all the /etc/resolv.conf files and query the BIND servers (Helen has been asking for this for months) and although we have done that in the DEN environment, apparently that hasn't been done on the LAX side yet.

Port Exhaustion Conclusions

- As a DBA you MUST understand how DNS is configured for every one of your databases
- As a DBA you MUST understand resolv.conf and monitor it for content and changes
- As a DBA you MUST educate DNS and System Admins about how to connect to a RAC cluster or a standby
- As a DBA, when troubleshooting connection issues, you MUST log in from an application server to identify what is actually going on ... you can't just FTP to the box

Case 5: Storage Storage Everywhere

and not a byte to to fill

Processing Stops and the NOC says

[Ticket] Commented: (1246816) mount points filled 100% on dc1laxdb01 and dc1laxdb03

Hi,

Two mounts got filled 100%, please add space as early as possible.
/u108 on dc1laxdb01
/export/home on dc1laxdb03

There are only datafiles in both mount points,

Thanks
Murphy

Ticket 1246816

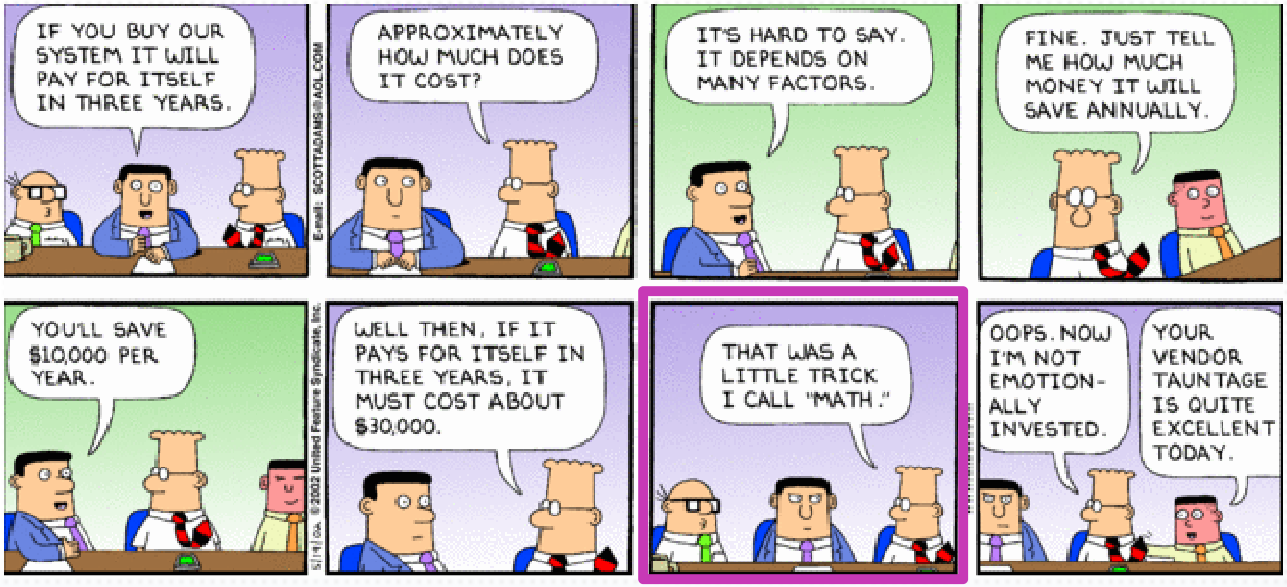
```
-bash-3.00$ df -h
Filesystem             size   used  avail capacity  Mounted on
/dev/md/dsk/d100        37G    11G    26G    29%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                    61G   2.1M    61G     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
sharefs                  0K     0K     0K     0%    /etc/dfs/sharetab
fd                       0K     0K     0K     0%    /dev/fd
/dev/md/dsk/d500        20G   4.6G    15G    24%    /var
swap                    62G   1.4G    61G     3%    /tmp
swap                    61G   142M    61G     1%    /var/run
/dev/dsk/c6t600601606AD11900E033B69AFA43DD11d0s2
                       115G    46G    68G    41%    /u01
/dev/md/dsk/d132        31G   2.2G    29G     8%    /var/crash
/dev/md/dsk/d60        9.8G   6.4G   3.3G    66%    /export/home
/dev/md/dsk/d402       422M   5.1M   374M     2%    /global/.devices/node@2
/dev/md/dsk/d404       481M   5.0M   428M     2%    /global/.devices/node@4
/dev/md/dsk/d401       415M    74M   299M    20%    /global/.devices/node@1
/dev/md/dsk/d403       481M   5.0M   428M     2%    /global/.devices/node@3
/dev/md/sf14/dsk/d112  4.2T   4.1T    34G   100%    /u112
/dev/md/sf14/dsk/d101  2.1T   2.0T    52G    98%    /u101
/dev/md/sf14/dsk/d109  2.1T   1.8T   239G    89%    /u109
/dev/md/sf14/dsk/d111  197G   3.5G   191G     2%    /u111
/dev/md/sf14/dsk/d100  2.1T   2.0T    31G    99%    /u100
/dev/md/sf14/dsk/d107  264G    73G   188G    28%    /u107
/dev/md/sf14/dsk/d102  1.0T  1005G    58G    95%    /u102
/dev/md/sf14/dsk/d106  264G    36G   225G    14%    /u106
/dev/md/sf14/dsk/d113  4.0T   3.6T   326G    92%    /u113
/dev/md/sf14/dsk/d110  3.0T   946G   2.0T    32%    /u110_arch
/dev/md/sf14/dsk/d104  2.0T   1.9T    37G    99%    /u104
/dev/md/sf14/dsk/d105  2.0T   2.0T   537M   100%    /u105
/dev/md/sf14/dsk/d108  2.0T   2.0T   2.0G   100%    /u108
/dev/md/sf14/dsk/d103  2.0T   1.9T    47G    98%    /u103
```

Storage Admin Tauntage: Let's Do Some Math

Total	Available
31	29
10	3
4200	34
2100	52
2100	239
197	191
2100	31
264	188
1000	58
264	225
4000	326
3000	2000
2000	37
2000	1
2000	2000
2000	47
27,266	**5,461**

The database is stopped because "they are out of space."

Yet 20% of the space allocated has never been used.



And That's Not Counting Free Space

```
SQL> SELECT file_name, tablespace_name
  2  FROM dba_data_files
  3  WHERE autoextensible = 'YES'
  4  ORDER BY 1;

FILE_NAME                                                  TABLESPACE_NAME
---------------------------------------------------------- ---------------
/u113/oradata/SF14/datafile/o1_mf_lob_01_8jlsmo05_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8jlst7ky_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8jlsx6fr_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8jlt035w_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8jlt34sd_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8rs5xndc_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8vdx8bps_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8vdx9r68_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8vdxc5ks_.dbf   LOB_01
/u113/oradata/SF14/datafile/o1_mf_lob_01_8vdxf9v1_.dbf   LOB_01

10 rows selected.


SQL> select sum(bytes)/1024/1024/1024 FREE_SPACE
  2  from dba_free_space
  3  where tablespace_name = 'LOB_01';

FREE_SPACE
----------
6166.08484
```

How Much Free Space Is There Really?

- Hard disk not fully occupied by data files
- Data files not fully occupied by segments
- Segments with extents above the high water mark
- Partially filled blocks
- Full blocks with empty space due to the PCTFREE setting
- LOB tablespace space occupied by undo data

Vacuum	Wasted
Not allocated for data files	5.5T
Freespace in LOB Tablespace	6.2T
Space in used blocks	2.3T
Total	14T (more than 50% of the 27.3T allocated)

- And yet the database is "out of space"

How Much Disk Space Really Isn't Real

- Disk space too small for another datafile
- Tablespaces with too little room for another extent
- Space that has been over-allocated due to storage virtualization

Storage Usage Conclusions

- Bytes that spin cost
  - money to purchase them
  - money to provide electricity to spin them
  - money to provide air conditioning to cool them
  - money to pay for the floor space they occupy
- Wasting space is wasting money
- Advanced and Hybrid Columnar Compression will not address the issues created by poor planning and usage
- You probably don't need pct_free set at 0%
- If you store BLOBs and CLOBs you need to know how much space is wasted storing undo
- Bigfile tablespaces are superior very often to SmallFiles
- There is almost no justification for lots of tablespaces and lots of datafiles: If you do it you must justify it

Case 6: UCS

and all the other blade servers we don't need and never asked for

But first ...

- Network stability is critical to Oracle DBAs
- If you have network issues you can waste staggering amounts of time proving the issue isn't the database
- I have worked for the last 10 months with Cisco UCS
  - ~10 databases stand-alone 11gR2
  - ~75 RAC Active-Active or Clusterware Active-Passive Failover
- The questions that need to be addressed are
  - What is the value of failover to a cluster?
  - What is the value of functioning network diagnostics?

VLANs and the Cluster Interconnect (1:2)

- It is essentially impossible do what is recommended in Oracle Support's "best practices" guidelines for RAC with blades: any blades from any vendor

RAC: Frequently Asked Questions (Doc ID 220970.1)

Cluster interconnect network separation can be satisfied either by using standalone, dedicated switches, which provide the highest degree of network isolation, or Virtual Local Area Networks defined on the Ethernet switch, which provide broadcast domain isolation between IP networks. VLANs are fully supported for Oracle Clusterware interconnect deployments. Partitioning the Ethernet switch with VLANs allows for:
- Sharing the same switch for private and public communication.
- Sharing the same switch for the private communication of more than one cluster.
- Sharing the same switch for private communication and shared storage access.

The following best practices should be followed:

The Cluster Interconnect VLAN must be on a non-routed IP subnet.

All Cluster Interconnect networks must be configured with non-routed IPs. The server-server communication should be single hop through the switch via the interconnect VLAN. There is no VLAN-VLAN communication.

Oracle recommends maintaining a 1:1 mapping of subnet to VLAN.

The most common VLAN deployments maintain a 1:1 mapping of subnet to VLAN. It is strongly recommended to avoid multi-subnet mapping to a single VLAN. Best practice recommends a single access VLAN port configured on the switch for the cluster interconnect VLAN. The server side network interface should have access to a single VLAN.

VLANs and the Cluster Interconnect (2:2)

- It is extremely difficult to troubleshoot interconnect issues with UCS as the tools for separating public, storage, and fusion interconnect packets do not exist

Troubleshooting gc block lost and Poor Network Performance in a RAC Environment (Doc ID 563566.1)

6. Interconnect LAN non-dedicated
Description: Shared public IP traffic and/or shared NAS IP traffic, configured on the interconnect LAN will result in degraded application performance, network congestion and, in extreme cases, global cache block loss.

Action: The interconnect/clusterware traffic should be on a dedicated LAN defined by a non-routed subnet. Interconnect traffic should be isolated to the adjacent switch(es), e.g. interconnect traffic should not extend beyond the access layer switch(es) to which the links are attached. The interconnect traffic should not be shared with public or NAS traffic. If Virtual LANs (VLANS) are used, the interconnect should be on a single, dedicated VLAN mapped to a dedicated, non-routed subnet, which is isolated from public or NAS traffic.

My Experience

- Blade servers, of which Cisco UCS are just one example, do not have sufficient independent network paths to avoid the networking becoming a single point of failure
- It is good when the public interface has a "keep alive" enabled but this is a fatal flaw for the cluster interconnect
- When different types of packets, public, storage, and interconnect are mixed low-level diagnostics are difficult if not impossible
- When different types of packets, public, storage, and interconnect are mixed the latency of one is the latency of all
- Traffic from any one blade impacts all blades

UCS Blade Server Conclusions

- Blade servers may be a good solution for application and web servers
- They may even be acceptable for stand-alone databases
- Blade servers are unsuitable when
  - High availability is the goal
  - RAC the technology for achieving it
  - Performance is critically important
  - You don't want to stay at work at night, on weekends, and holidays troubleshooting repeated unexplained failures

Case 7: 5010 <> 7010

Things can break even when nothing is broken

What You Can't See Matters

- 6 Node RAC Cluster ... each node is an Sun M9000
- Storage is 3 clustered EMC VMax arrays = 1.25PB
- Public is bonded and redundant 10gEth
- The initial fusion interconnect is a single Cisco 5010
- The new interconnect is two bonded Cisco 7010s

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
OPM01P	782247420	opm01p6	6	18-Aug-10 21:08	11.1.0.7.0	YES

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
usp9004b	Solaris[tm] OE (64-bit)	128	64	16	503.16

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	7037	15-Sep-10 13:00:18	406	7.5
End Snap:	7038	15-Sep-10 14:01:28	318	8.5
Elapsed:		61.17 (mins)		
DB Time:		6,076.88 (mins)		

- What could possibly go wrong?

Every Node Starts If It Is First

- Shutdown the RAC Cluster
- Pull 10gEth from Cisco 5010 and plug into Cisco 7010
- Start any one node of the cluster
- No other node joins the cluster: the order doesn't matter
- Plug cables back into the 5010 and all is well
- Cisco engineers verify the 7010 is in perfect condition
- Repeat the above steps ... result is the same no matter which node is started first ... no other node can join the cluster
- Repeat numerous times with different start orders ... the result is always the same

What Is Happening?

- The first node started registers itself with the voting disk and knows no other nodes have started
- The second node, no matter which one is second, registers with the voting disk, sees that another node is in the cluster and tries to communicate with it
- The brand new "perfect" Cisco 7010 rejects every packet sent and the failure to communicate causes the second node to shoot itself in the head
- The question is: Why?

Cisco 5010 to 7010 Migration Conclusions

- Cisco introduced a new error checking algorithm in the 7010s when they were initially released
- The algorithm rejected every RAC cluster interconnect packet as corrupt
- A resend request was made and the resent packets also failed to arrive at their destination creating a storm
- It is possible for everything to be working perfectly and yet for the full, integrated, system to fail
- For once the network admins were correct that the network was not to blame ... and yet it was, as is so often the case, the source of the failure

Case 8: RAC Is RAC

Server Manager usage is not optional

Ticket 108 (1:2)

- RCA Request for DC20 | Database | All databases are down due to memory issue and its 100% full. Here is the alert log

```
system name: Linux
Node name: oras1n1.lux20.morgan.priv
Release: 2.6.18-274.el5
Version: #1 SMP Mon Jul 25 13:17:49 EDT 2011
Machine: x86_64
Redo thread mounted by this instance: 1
Oracle process number: 0
Unix process pid: 32402, image: oracle@oras1n1.lux20.morgan.priv (J000)


*** 2013-07-04 03:51:11.919
Unexpected error 27140 in job slave process
ORA-27140: attach to post/wait facility failed
ORA-27300: OS system dependent operation:invalid_egid failed with status: 1
ORA-27301: OS failure message: Operation not permitted
ORA-27302: failure occurred at: skgpwinit6
ORA-27303: additional information: startup egid = 1001 (oinstall), current egid = 1003 (asmadmin)

Errors in file /app/oracle/base/diag/rdbms/dc20sce11/DC20SCE11/trace/DC20SCE11_j000_32402.trc:
ORA-27140: attach to post/wait facility failed
ORA-27300: OS system dependent operation:invalid_egid failed with status: 1
ORA-27301: OS failure message: Operation not permitted
ORA-27302: failure occurred at: skgpwinit6
ORA-27303: additional information: startup egid = 1001 (oinstall), current egid = 1003 (asmadmin)

And current status of memory usage:
 oracle@oras1n1.lux20.morgan.priv[DC20SCE11]$ free -g
             total used free shared buffers cached
Mem: 141 140 1 0 0 66
-/+ buffers/cache: 73 67
Swap: 31 0 31
```

Ticket 108 (2:2)

Root Cause:

A review of Oracle Binary in oras1n1 revealed that Oracle Databases were started by user “oracle” and at that point of time the ORACLE_HOME/bin/oracle executable group was “oinstall”. The ORACLE_HOME/bin/oracle executable group was accidentally changed to “asmadmin”, due to a known Oracle bug. Due to this bug, cluster nodes originally started with Server Control must always be started with Server Control and, if started with SQL*Plus, can produce the result observed.

Corrective Action:

Need to change the group of executable “oracle” to “oinstall” for all the database homes, if they have been modified. The bug hit has been acknowledged by Oracle and, at least in theory, should be fixed in version 12cR1 and above. Further improvements will be tracked by CSI via a Corrective Measure (CM).

Server Manager Is Not Optional Conclusions

- DBAs MUST have training to successfully navigate our increasingly complex set of tools
- You can not run RAC successfully without knowing more than just syntax
- Oracle created UNDO tablespaces because DBAs didn't understand rollback segments
- Oracle created SCAN IP addresses because DBAs didn't understand how to create and manage services
- Hardly any DBAs build ASM diskgroups with ATTRIBUTE (HOT MIRRORHOT)
- Most DataGuard failures I've seen have been the result of not using DataGuard broker
- Do not expect good things to happen if you start and stop RAC clusters with SQL*Plus

ERROR at line 1:
ORA-00028: your session has been killed

Contact: dmorgan@forsythe.com
+1 206-669-2949
Skype: damorgan11g

Thank you