



BASIC DEBUGGING TECHNIQUES

Agenda

- Understanding Server Working Directory (.swd)

- Enabling debugging

- Retrieving Current Vovserver Summary

- Investigating Sick Slaves

- Summary & Quiz



.SWD Structure

- Each Vovproject has an associated server working directory (*projectName.swd*)
- How to find out current server working directory
 - nc cmd vovserverdir -p .

```
[rtdamgr@dexrt01 ~]$ nc cmd vovserverdir -p .
/sw/rtda/vnc/vnc201901.swd/
[rtdamgr@dexrt01 ~]$
```

- What are the contents of .swd?

```
[rtdamgr@dexrt01 vnc201901.swd]$ ls
autostart          crontab.dexrt01      equiv.tcl          html               resources.tcl      setup.tcl          voveventstatd    vovpreemptd      waves
cgiconfig.tcl     crontab.dexrt01.old.1  exclude.tcl       jobclass           saved              slaves.tcl        vovlad           vovresourced
cgienv            data                 fairshare         journals           scripts            staging           vovnetbatchd    vovserver.pid
cleanup.config.tcl db                   gif              logs               security_info     tasks            vovnginxd       vovtriggerd
config            dump                 gui               open_a_vovconsole.csh  security.tcl      trace.db          vovnotifyd      vovvxd
cron.csh          environments         gui.tcl           policy.tcl         serverinfo.tcl    vovdbd           vovperiodicd    vovzipdir
```

- Some important directories to consider
 - Logs, journals, jobclass, fairshare, data, daemon directories and their individual logs, trace.db (persistent and crash recovery representation)



Enabling debugging

- Enable debugging for various components in webUI
- nc_host:nc_port/admin?page=debug
 - Please be advised that debug log files can get very large quickly and remember to turn off debugging once you are finished with it.
 - Sometimes even after turning on debugging, one may not see debug output in log files.
 - One needs to increase the default log verbosity of the feature.
e.g.
nc cmd vovservermgr config preemption.log.verbosity 7
- Now you can follow specific logs in .swd and take necessary actions.

Server Debugging

This page is meant to be used by expert developers.

Turning debugging ON causes the log file to become very large. **We recommend you always leave all debugging flags OFF.**

	Flag Name	Value (click to toggle)
All Flags		
1	VovBuffer	OFF
2	CompressBuffer	OFF
3	Bzocket	OFF
4	Clients	OFF
5	Registry	OFF
6	MmChunks	OFF
7	HalfAtt	OFF
8	AttachmentLists	OFF
9	Events	OFF
10	EventFilter	OFF
11	SubServerEvents	OFF
12	NotifySet	OFF
13	SafeThreads	OFF
14	SubServer	OFF
15	ServerConfig	OFF
16	ListElements	OFF
17	LogicalNames	OFF
18	NFS	OFF
19	Host-DNS	OFF
20	Features	OFF



Retrieving Current Vovserver Summary

- Following command shows information and summary about current Vovserver.

- nc cmd vsi

```
[rtdamgr@dexrt01 ~]$ nc cmd vsi
```

- Things to watch out for

- Alerts

- Jobs and files

- Size (memory)

- Files

- Buckets

- Scheduler time

- Overloaded slaves

- Zap away bad nodes

- nc cmd vsz *or* nc clean -zap

- (can be an expensive operation)

```
Vov Server Information - 10/28/2019 11:03:44
```

```
vnc201901@dexrt01:10645 | URL: http://dexrt01:10645
```

```
-----
Jobs:                1,000 | Workload:
Files:               1,002 | - running:                18
Sets:                22 | - queued:                 969
Retraces:           0 | - done:                   13
                    | - failed:                  0
```

```
-----
Slaves:              4 | Buckets:                   1
- ready:             1 | Duration:                  1h20m
- ovrlld:            3 | SchedulerTime:             0.00s
Slots:              52 |
```

```
-----
TotalResources:     328 | Pid:                       10707
                    | Saved:                     1h29m ago
                    | Size:                      47.00MB
                    | TimeTolerance:             3s
```

```
[rtdamgr@dexrt01 ~]$ █
```



Investigating Sick Slaves

- The Vovserver marks a vovslave SICK when it has not received the vovslave's heartbeat message for three consecutive update cycles.
- Stop and start the slave with *nc cmd vovsavemgr stop/start slaveName*
- Check the process tree – *pstree –cupal*
- Check network information – *netstat –a (open communication)*
 - Check for a process running on specific port (may require root privileges)
netstat –tulnp | grep vov
- Is the vovslave/vovslaveroot process stuck
 - Find the process ID on vovslave – *ps –ef | grep vov*
 - Attach the process(on Linux) to *strace –Ttt –p PID*
 - Alternatively use *vovpstack PID*
- Often output of above commands shows if the process is stuck in certain section and remedial actions can be taken (e.g. waiting for filer, listening on port etc.)



Summary & Quiz

- Understanding server working directory (.swd) and its contents.
 - Enabling debug information for various components
 - Server status quick summary and identifying hotspots
 - Cleaning up empty sets and buckets (vovforget)
 - Using system tool to investigate sick slaves
-
- How to identify and daemon directory and logs associated with it?
 - Which system tools (Linux) can one use to investigate sick slaves or processes?







JOBCLASSES BEST PRACTICE



Agenda

- ❑ Job Class equivalents using direct NC flags

- ❑ What should we always add to a jobclass?

- ❑ Modifying the entire queue –

1. Default jobclass
 2. vnc_policy.tcl
 3. vncrun.config.tcl
-

- ❑ Reviewing jobclass reports

- ❑ Summary



JOB CLASS EQUIVALENTS USING DIRECT NC FLAGS



Job Class equivalents using direct NC flags

- A Job Class is the best shortcut you can have to summarize the commonly used NC flags for similar job types.
- The Job Class does not add new flags to NC , all of its features have their equivalent commands within the NC system.
- We would not recommend to define more than ~50-100 classes.
- Reading is done from left to right.



WHAT SHOULD WE ALWAYS ADD TO A JOBCLASS



What should we always add to a jobclass?

- For each job class define:
 - Cores/RAM
 - Env necessary for the job – named env are recommended
 - Autokill – please use, we have a notification script to share.
 - Group, can be a variable [needed for fairshare allocations], example: `set VOV_JOB_DESC(group) "/proj/$proj"`
 - Slavelist
 - JPP
 - Profile for farm under 3K – can stay on all the time
 - Limits - define only if needed, we don't encourage limits
 - Interactive flag if needed



QUIZ

**WHY IS SLAVELIST A BETTER MECHANISM
THAN SLAVE RESERVATION?**



MODIFYING THE ENTIRE QUEUE



Default jobclass

- 3 options to a flag in the jobclass setup: use, don't use, don't care.
- If your job has a “don't care” flag in it, it will read the settings from the default job class (if configured).



vnc_policy.tcl

- To use, create this file in vnc.swd.
- Only use if you know you need it.
- vnc_policy.tcl kicks in at the end of job submission to parse through the entire command user punched in.
- Example:

```
proc VncPolicyValidateResources { resList } {  
    #  
    # This policy sends tharas jobs to vovslave hosts offering 'tharas_host'  
    # and keeps other kinds of jobs off those hosts  
    #  
    catch {  
        set jtool [VncPolicyGetJobInfo tool]  
        if { "$jtool" == "tharas" } {  
            lappend_no_dup resList tharas_host  
        } else {  
            lappend_no_dup resList "!tharas_host"  
        }  
    }  
    return $resList  
}
```



vncrun.config.tcl

- vncrun.config.tcl works at the beginning of job submission, ex nc run = nc run -e ...etc
- Can be used as a global class file definition for the entire queue.
- By default will work on all queue in the installation.
- Location: \$VOVDIR/../../local/



REVIEWING JOBCLASS REPORTS



Reviewing jobclass reports

- Project Home > Sets > Browse > Class
- Project Home > Job Queue > Job Plots



SUMMARY



QUIZ

**MY JOB RECEIVED AUTOKILL 1H,
IT ISN'T DEFINED IN MY JOBCCLASS OR MY
COMMAND.
WHAT 2 THINGS MAY EXPLAIN IT?**



Summary

- Jobclasses is one of the most important tools we have as Admins, to configure our system.
- For a fully functional engineering environment, Jobclasses will help you guide the users easily, and create a common language to discuss job types.







USING VARIOUS INTERFACES AND DOCUMENTATION EFFECTIVELY
& UNDERSTANDING VOV SUBSYSTEM & TOOLKIT



Agenda

- ❑ Command Line Interface (CLI)
 - ❑ Interacting With Vovserver
 - ❑ Interacting With Altair Accelerator (NC) Vovserver
 - ❑ VOV Project WebUI
 - ❑ Searching HTML Documentation
 - ❑ Frequently Asked Questions & How To
 - ❑ VOV Subsystem
 - ❑ VOV Toolkit
 - ❑ Summary & Quiz
-



Command Line Interface (CLI)

- Altair Accelerator Package's command line interface is self-discovering
- When invoked without any options, it displays inbuilt help and few typical examples.
- e.g.

```
[rtdamgr@dexrt01 ~]$ nc info

vnc: Usage Message
NC INFO:
  Get information about a specific job or list of jobs.
USAGE:
  % nc info [options] <jobId> ...
OPTIONS:
  -h                -- Show this message.
  -v                -- Increase verbosity.
  -l                -- Show the log file (actually, it shows all outputs).
  -e                -- Show the environment name, or contents if a snapshot.
  -c                -- Show slave compatibility table (which slave can run a job).
  -sc               -- Show slave compatibility in normal output.
  -dep              -- Show job dependencies.
  -J <jobname>     -- Show the jobs with given name.

EXAMPLES:
  % nc info 00123456      -- Show info about specific job.
  % nc info !            -- Show info about most recent job in current dir.
  % nc info -l 12345     -- Show log file(s) of job.
  % nc info -J MyJob     -- Show info about all jobs called "MyJob".
  % nc info -sc 0012345  -- If job is Scheduled, also show the summary of slave compatibility.

[rtdamgr@dexrt01 ~]$
```



Interacting With Vovserver

- Various commands can be invoked directly on command prompt when the VOV project is enabled.
- e.g. Altair Monitor or FlowTracer
- vovsh* is a major component of Altair Accelerator Package products.
- Can run Tcl scripts and extensions provided by Altair.

```
[rtdamgr@dexrt01 ~]$ vovproject enable licmon201901
vovproject 10/28/2019 14:46:56: message: Enabling project 'licmon201901'...
dexrt01 licmon201901@dexrt01 [DEFAULT] 1086 > vsi

Vov Server Information - 10/28/2019 14:47:00

licmon201901@dexrt01:5557          | URL: http://dexrt01:5555
-----
Jobs:                             17 | Workload:
Files:                             23 | - running:                2
Sets:                              31 | - queued:                 0
Retraces:                          0 | - done:                   13
                                   | - failed:                 2
-----
Slaves:                            5 | Buckets:                  0
  - ready:                         4 | Duration:                 0s
  - full:                           1 | SchedulerTime:           0.00s
Slots:                             11 |
-----
TotalResources:                    15 | Pid:                      3495
                                   | Saved:                    1h15m ago
                                   | Size:                     43.00MB
-----

dexrt01 licmon201901@dexrt01 [DEFAULT] 1087 > █
```



Interacting With Altair Accelerator (NC) Vovserver

- A special case for Altair Accelerator (NC) or Accelerator Plus (WX)
- Interact with Vovserver using *nc cmd/wx cmd*
- The commands execute in context of Vovserver running under NC/WX

```
[rtdamgr@dexrt01 ~]$ nc cmd vovshow -buckets
      ID   JOBS  PRI  FSGROUP      USER      AGE  RANK  WAITING FOR...
000064997   971   4  /time/users  rtdamgr     0s    0  HW
000065602   299   4  /time/users  rtdamgr     0s    0  HW
[rtdamgr@dexrt01 ~]$
```

- Simple command line utilities and complex scripts can also be executed with above approach.

```
[rtdamgr@dexrt01 ~]$ nc cmd vovsh -x 'vtk_generic_get project P; puts $P(logfile)'
/sw/rtda/vnc/vnc201901.swd/logs/server.2019.10.28_100443.log
[rtdamgr@dexrt01 ~]$
[rtdamgr@dexrt01 ~]$ nc cmd vovsh -f foo.tcl
/sw/rtda/vnc/vnc201901.swd/logs/server.2019.10.28_100443.log
[rtdamgr@dexrt01 ~]$
```



VOV Project WebUI

- Along with command line interface, many configuration options can be retrieved or set from webUI.
- e.g. General info, serve config, database config etc.

Project Home > Admin

SmartSearch

Actions

- Save
- Reread
- Sanity
- Shut Down

Administration

- Alerts
- License
- System Recovery
- Database
- Daemons
- Environments
- Periodic Jobs
- Users
- Who

General Information

Project Name	vnc201901
Product Name	nc
Server Type	primary
Host	dextr01
Port	10645
Web Port	10645
Read-Only Port	0 OFF
Working Directory	/sw/rtda/vnc
Server Version	Linux64/2019 Update 1, Build 72579 built:Jun 14 2019 12:49:08
Server Size	47,206,400 bytes (estimated)
Server Delay	1m04s
Server Capacity	2500
Automatic Shutdown	never

Project Home > Admin > Server Configuration Parameters

SmartSearch

Parameter	Value	Config File	Type	Default	Range	Description
1 acl.default_jobs_leader	VIEW,EXISTS,CHOWN	policy.tcl	string	VIEW,EXISTS,CHOWN		Default ACL for FlowTracer jobs assigned
2 alerts_max	50	policy.tcl	integer	50	[5--1000]	Maximum number of alerts. If more alerts are received, the oldest alerts are deleted. If nonzero, request the vovserver to dump alerts when the number of alerts was zero.
3 allowcoredump	1	policy.tcl	boolean	0	0,1	
4 allowUidForSecurityFile	0	policy.tcl	integer	0	[0--2147483648]	Additional UID that can own the security file.
5 autoForgetFailed	2400h	policy.tcl	timespec	2 days	[5m,infinity]	If the autoForget bit is set on a job, the specified time.
6 autoForgetOthers	2400h	policy.tcl	timespec	2 days	[5m,infinity]	If the autoForget bit is set on a job, the specified time.
7 autoForgetRemoveLogs	0	policy.tcl	boolean	0	0/1	Try to remove the log files associated with the specified time.
8 autoForgetValid	1h00m	policy.tcl	timespec	1 hour	[5m,infinity]	If the autoForget bit is set on a job, the specified time.
9 autoLogout	8h00m	policy.tcl	timespec	4 hours	[10m,infinity]	The period of inactivity after which the minimum value for this parameter is 10 minutes.
10 autoRescheduleCount	4	policy.tcl	integer	4	0-10	Specifies the maximum number of reschedule attempts.
11 autoRescheduleOnNewHost	1	policy.tcl	boolean	1	0/1	Controls whether auto-rescheduling will run on a different slave on the same host. Does not apply to auto-rescheduling base.
12 autoRescheduleThreshold	0s	policy.tcl	timespec	0s	0s-60s	This parameter deals with jobs that fail and are automatically resubmitted at high priority. If the value is 0, the job will be placed on a different slave. A maximum of 4 auto-reschedules are permitted. autoRescheduleCount parameter. Host/slave behavior can be controlled via autoRescheduleThreshold parameter. Does not apply to auto-rescheduling base.
13 autoShutdown	0s	policy.tcl	timespec	2w	[15m,infinity]	A time specification. This parameter controls the time after which the vovserver will shut down. If the value is 0, the server will shut down immediately. The minimum acceptable shutdown period is 15 minutes.
14 blackholeDetection	0	policy.tcl	boolean	0	0/1	If set, activates black-hole detection.

Database Configuration

Use the page to configure and control the database, check database status, and review database statistics.

Database Location

Host: Refer to the [documentation](#) if configuring the database to run on a remote host.

Port: Port of the database to listen for connections. By default, the system initialized a random port to use.

Path: Path to the database storage location as seen by the specified host. It is highly recommended to use a path that is located on the specified host's local disk for performance and reliability reasons. It is also recommended to include this product's instance name (vnc201901) in the path to help identify the directory's purpose.

Database Control

ON

Database Tasks


These tasks require the **vovbdb daemon**.
Daemon status: running

Enable Data Loader



Searching HTML Documentation

- HTML documentation deck can be found under <http://ncHost:ncport/cgi/bookshelf.cgi>
- Search keywords in HTML documentation which queries all the available documentation.



MATCH CASE:

ALTAIR ACCELERATOR™ PACKAGE DOCUMENTATION BOOKSHELF

<p>ALTAIR ACCELERATOR PACKAGE -----</p> <ul style="list-style-type: none"> · ALTAIR ACCELERATOR PACKAGE INSTALLATION PDF · ALTAIR ACCELERATOR PACKAGE RELEASE NOTE PDF · GLOBAL INDEX <p>VOV SUBSYSTEM THE UNDERPINNINGS OF ALL PRODUCTS -----</p> <ul style="list-style-type: none"> · VOV™ SUBSYSTEM ADMINISTRATOR GUIDE PDF · VOV™ SUBSYSTEM REFERENCE MANUAL PDF <p>ALTAIR ACCELERATOR FAST SCHEDULING -----</p> <ul style="list-style-type: none"> · ALTAIR ACCELERATOR™ ADMINISTRATOR GUIDE PDF · ALTAIR ACCELERATOR™ USER GUIDE PDF · ALTAIR ACCELERATOR™ TRAINING PDF <p>ALTAIR ACCELERATOR PLUS HIERARCHICAL SCHEDULING -----</p> <ul style="list-style-type: none"> · ALTAIR ACCELERATOR PLUS™ ADMINISTRATOR GUIDE PDF · ALTAIR ACCELERATOR PLUS™ USER GUIDE PDF 	<p>ALTAIR MONITOR CURRENT AND HISTORICAL LICENSE UTILIZATION -----</p> <ul style="list-style-type: none"> · ALTAIR MONITOR™ ADMINISTRATOR GUIDE PDF · ALTAIR MONITOR™ USER GUIDE PDF <p>ALTAIR ALLOCATOR LICENSE ALLOCATION ACROSS ENTERPRISE -----</p> <ul style="list-style-type: none"> · ALTAIR ALLOCATOR™ ADMINISTRATOR GUIDE PDF <p>ALTAIR FLOWTRACER DESIGN FLOW MANAGEMENT -----</p> <ul style="list-style-type: none"> · ALTAIR FLOWTRACER™ USER MANUAL PDF · ALTAIR FLOWTRACER™ DEVELOPER MANUAL PDF · ALTAIR FLOWTRACER™ ADMINISTRATOR MANUAL PDF · EDA TOOL INTEGRATION GUIDE PDF · THE ART OF FLOWS PDF <p>ALTAIR HERO HARDWARE EMULATOR RESOURCE OPTIMIZER -----</p> <ul style="list-style-type: none"> · ALTAIR HERO™ MANUAL PDF
--	--



Command Line Documentation

- Alternatively to HTML, one can search documentation on command line using `vovdoc` utility

```
[rtdamgr@dexrt01 ~]$ vovdoc

vovdoc: Usage Message

DESCRIPTION:
  The utility vovdoc scans the files in $VOVDIR/doc/txt/*/*
  for documentation that matches the words provided on
  the command line.

SYNOPSIS:
  % vovdoc [OPTIONS] word ...

OPTIONS:
  -h          -- This help
  -v          -- Increase verbosity
  -n <N>     -- How many documents to show (default 1)
  -s          -- Show keywords with <<<WORD>>>
  -test      -- Test installation of txt docs.

EXAMPLES:
  % vovdoc vovbuild
  % vovdoc vovid
  % vovdoc vovid -s
  % vovdoc exit status

[rtdamgr@dexrt01 ~]$
```



Frequently Asked Questions & How To

- Over the years we have gathered some frequently asked questions from users and publish them in article format on our website
- This is a collection of FAQs, How Tos and some utility scripts which can be adapted to your environment
- <https://aap-support.pbsworks.com/>



VOV Subsystem

- VOV Subsystem consists of rich set of programs that can be run in shell command line.
- They are grouped in categories as:
 - General – *vsi, vsz, vsz, vovconsole etc.*
 - User – *vovbrowser, vovsh, vovshow, vovversion etc.*
 - Administration – *vovserver, vovenv, vovserverdir, vovslavemgr etc.*
 - Other – *vovprop, vovshowconnection, vovmemtime and many more*
- Handy utility to simulate workload – use *vovmemtime*
- *vovmemtime* occupies specified block of memory and high CPU load for given duration
- Additionally with recursion levels, it spawns child processes with same parameters to track memory usage.
- Usage: *vovmemtime mem_inMB time_in_seconds recursion_level*



VOV Toolkit

- VOV Toolkit (vtk) provide a rich set of procedures to interact with Vovserver for configuration and retrieval information.
- VTK procedures can be used in isolation or invoked from Tcl scripts.
- VTK procedures are also used in many project configuration files and job classes.
- e.g. Set maximum number of clients for notifications.

```
[rtdamgr@dexrt01 ~]$ nc cmd vovsh -x 'vtk_server_config maxNotifyClients 400'  
[rtdamgr@dexrt01 ~]$ █
```

- e.g. In the event of server crash allow the core dump to be examined later

```
[rtdamgr@dexrt01 ~]$ nc cmd vovsh -x 'vtk_server_config allowcoredump 1'  
[rtdamgr@dexrt01 ~]$ █
```

- Demo : Using vtk commands in script.



Summary & Quiz

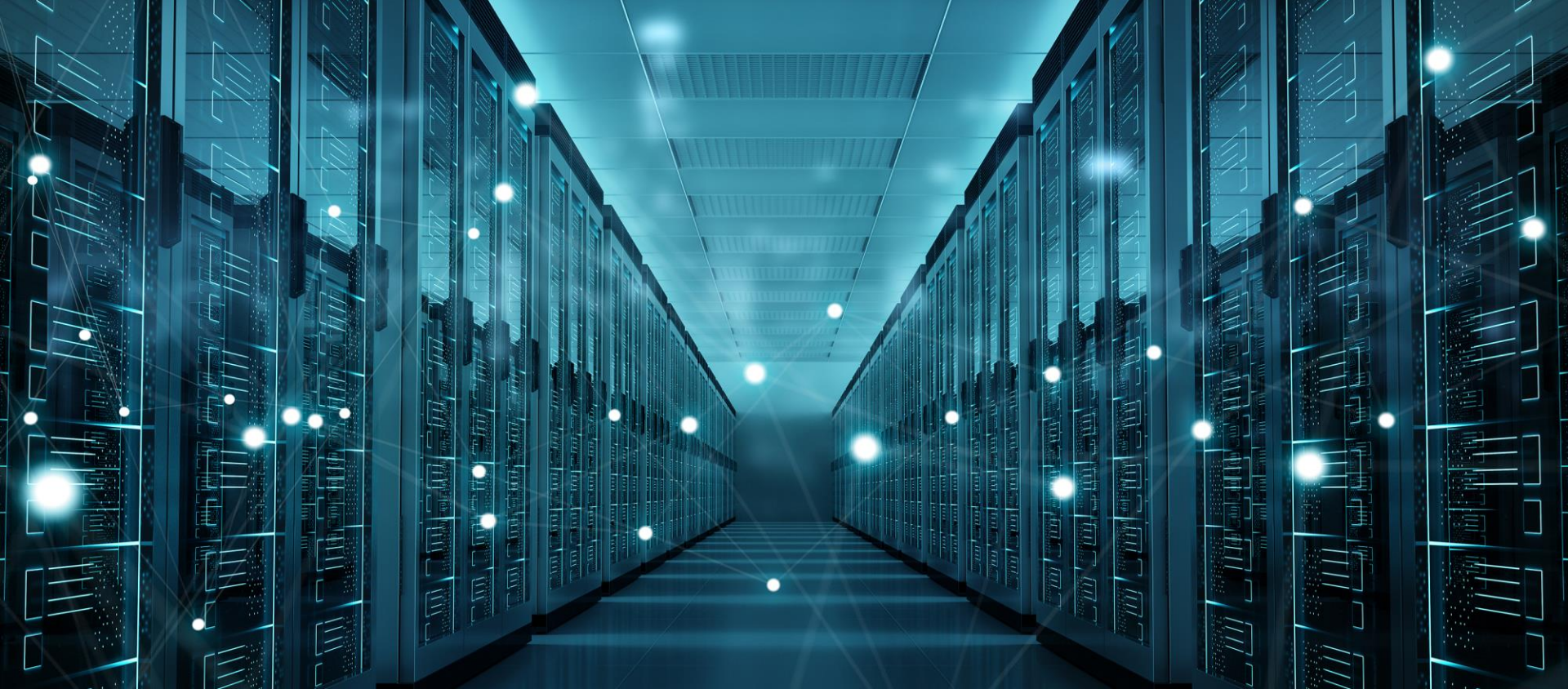
- Using self-discovering documentation for quick look and examples
- Using command line and web interface to interact with vovserver
- Using command line documentation with *vovdoc*
- Understanding VOV subsystem and toolkit (vtk)
- Quick demo of VTK in a script.

Question

- Where can one find server configuration parameters and their values?
- How can one configure NC Vovserver on command line?
- I want to keep failed jobs in memory for longer duration than default (2 days). Which command can I use?







PREEMPTION BEST PRACTICE

Agenda

- Intro to FairShare

- Fairshare setup

1. CLI utility
 2. saved file
-

- What is preemption?

- Writing preemption rules

- Debugging preemption

- Summary



INTRO TO FAIRSHARE



Concept of Fairshare

Fairshare table
[To get the target share for each group]

Fairshare window
[To get the actual share each group received during the last window cycle]

Fairshare mechanism

under-allocated
Target > Actual

over-allocated
Target < Actual

Boost workload of under-allocated

Reduce workload of over-allocated



FAIRSHARE SETUP



Fairshare setup

- Do not use Time and System hierarchies as your fairshare branches. Create new branches.
- Vovfsgroup -h
- Can create fairshare groups by project, class, by software, and more.
- Fair share is the built-in recommended mechanism that we recommend that you use to prioritize your resources.



QUIZ

WHAT ARE THE 2 INPUTS NECESSARY FOR THE FAIRSHARE MECHANISM TO WORK?



WHAT IS PREEMPTION?



What is preemption?

- Preemption is the process of helping "important jobs" that are queued by either revoking resources from one or more running job or by reserving resources.
- Preemption is normally triggered by queued jobs, but it can also be triggered a manual request from the user.
- Preemption may require killing or suspending running jobs.



WRITING PREEMPTION RULES



Writing preemption rules

- First be aware that we can do manual preemption using “nc preempt”, for example:
nc preempt -v -v -v -manualresume -method SIGTSTP+LMREMOVE jobid

- There are two ways to manage preemption rules:
 1. Rules can be created and edited using the web interface.
 2. Rules can be defined in a Tcl-syntax configuration file vnc.swd/vovpreemptd/config.tcl.

- Examples: preemption for HW, preemption for SW.



DEBUGGING PREEMPTION



Debugging preemption

- Server_preemption.log can be enabled:

```
vovsh -x 'vtk_server_config rotate_server_preemption_log 1'
```



QUIZ

**WHAT 2 INGREDIENTS WILL AFFECT THE
RECIPE FOR SUCCESSFULLY SUSPEND-
RESUME A JOB?**

**AND WHAT PREEMPTION METHOD SHOULD WE
USE IF WE DON'T HAVE A WORKING RECIPE?**



SUMMARY



Summary

- Always remember that fairshare is the scheduler mechanism to dispatch according to your preferences.
- When using “kill & resubmit” the re-queued job will obey to the scheduler fairshare rankings.
- When the scheduler is preempting, it is not scheduling.
Only use preemption if you know you need it.
- Not more than a few rules, and you can tune time cycle when to shoot the rules.







ADVANCED TOPICS



Agenda

- RAM Sentry
 - Writing Custom Health Checks
 - Automatic Cleaning Of Job Processes
 - Altair Accelerator Database Overview
 - Summary & Quiz
-



RAM Sentry

- RAM sentry guards against processes from being killed by kernel OOM killer.
- It performs safety measures to prevent slaves memory being saturated.
- When activated, the process OTHER THAN largest memory footprint are suspended.
- When RAM sentry is active, slave is suspended and does not accept more jobs.
- To activate on all slaves before starting, use *vnc.swd/setup.tcl*
 - *setenv VOV_RAM_SENTRY 1*
- To change on a live slave
 - *nc cmd vovslavemgr configure -ramentry 1 hostID*



Writing Custom Health Checks

- Inbuilt health check notifications can be enabled/disabled from webUI.
- Custom health check procedures should always begin with “*doTestHealth*”
- Use reference inbuilt script “*vovhealthlib.tcl*”
 - *\$VOVDIR/local/vovhealthlib.tcl*
- Write local health in a file called “*vovhealthlib.tcl*”
- Place the file in *vnc.swd/vovnotifyd* daemon directory
- Register the health check in *config.tcl* for *vovnotifyd*
 - *vnc.swd/vovnotifyd/config.tcl*
- New health check will appear in webUI.

Notification Configuration

Use the forms below to configure the notification system.

Health Checks | [SMTP Configuration](#) | [E-Mail Maps](#)

Procedure	Status	Frequency		Recipients	Actions	
		Check	Mail		Edit	Disable
AllJobsFailedOnHost	✓	10m00s	1d00h	@OWNER@	Edit	Disable
CheckAlerts	✓	10m00s	1d00h	@OWNER@	Edit	Disable
CheckDownSlaves	✓	10m00s	1d00h	@OWNER@	Edit	Disable
CheckJobsReqstRam	✓	10m00s	1d00h	@OWNER@	Edit	Disable
CheckSlaveset	✓	10m00s	1d00h	@OWNER@	Edit	Disable
CheckVendorLicenseExpiration	✓	10m00s	1d00h	@OWNER@	Edit	Disable
Daemons	✓	10m00s	1d00h	@OWNER@	Edit	Disable
FailoverServerCandidates	✓	10m00s	1d00h	@OWNER@	Edit	Disable
FalseLicenseUsage	✓	10m00s	1d00h	@OWNER@	Edit	Disable
JobStuck	✓	10m00s	1d00h	@OWNER@	Edit	Disable
JobsWaitingDefaultJobclass	✓	10m00s	1d00h	@OWNER@	Edit	Disable
JobsWaitingForTooLong	✓	10m00s	1d00h	@OWNER@	Edit	Disable
KillBullyJobs	✓	10m00s	1d00h	@OWNER@	Edit	Disable
LongJobs	✓	10m00s	1d00h	@OWNER@	Edit	Disable
RamSentry	✓	10m00s	1d00h	@OWNER@	Edit	Disable
ServerDiskSpace	✓	10m00s	1d00h	@OWNER@	Edit	Disable
ServerSize	✓	10m00s	1d00h	@OWNER@	Edit	Disable
TooManyFailures	✓	10m00s	1d00h	@OWNER@	Edit	Disable
TooManyOutOfQueueJobs	✓	10m00s	1d00h	@OWNER@	Edit	Disable



Automatic Cleaning Of Job Processes

- Some tools leave child processes after main parent process exits.
- This may cause vovslave to overload and stop accepting jobs.
- Accelerator can be configured to enable automatic cleanup of such processes.
- Scan VOV_JOBID env var for each process and match with recently ended processes.
- To enable this feature
 - Open *vnc.swd/policy.tcl*
 - *set config(slave.childProcessCleanup) 1*
 - *nc cmd vovproject reread*
- To confirm if this feature is enabled or not
 - *nc cmd vovselect param.slave.childProcessCleanup from server*
- For orphan child processes to be killed, corresponding message will appear in slave log.



Altair Accelerator Database Overview

- Vovdbd – database daemon triggers periodic tasks in background
- Can be configured by command line or webUI.
- Database based on PostgreSQL™ (9.4 & 9.6)
- *vovdb_util* is a powerful command line tool to configure and control database
- Additionally jobs database can be manually loaded using *vovsql_load_jobs*
- If updating AAP from a version prior to 2015.09, it is possible to pass the **-pre201509** option to ***vovdb_util dump*** to generate data files from the old database.
- The script *jobplots.cgi*, accessible via the URL *http://HOST:PORT/cgi/jobplots.cgi*, can generate plots of jobs over a period of time.



Summary & Quiz

- Enabling RAM sentry to guard against slave memory saturation
- Writing and customizing health checks
- Automatic cleaning of orphan job processes.
- Accelerator database overview and writing custom queries.

Question

- How does automatic cleaning of processes work?
- I have an older AAP(NC) database. How do I import it to latest version?



