

Space Details

Key:	CoreData
Name:	CoreData
Description:	Information about MBARI Data Users Committee managed core data streams including ROVNAV, ROVCTD, AUVCTD, Mooring.
Creator (Creation Date):	mccann (Jun 08, 2007)
Last Modifier (Mod. Date):	mccann (Feb 05, 2009)

Available Pages

- MBARI Core Data 
 - ArcMapNav Demonstration
 - AUVCTD Processing Data Flow Diagram
 - Contacts
 - DataAccessMeeting25June2008
 - DataFlowPage
 - Editing Instructions
 - History of Navigation Editing
 - Known Navigation Editing Issues
 - Current Status of navigation Editing
 - Misc. Notes
 - Navigation Editing Instructions
 - Additional hints and tips for Navediting
 - Navigation to be reviewed for re-edit
 - navStatsNotes
 - PostgreSQL Server kraken setup
 - PostgreSQL-9.1 Server Tuning
 - Recently re-edited and to-be-reviewed navigation files
 - Recently re-edited and to-be-reviewed navigation files*
 - ROVGISUpgradeMeeting03Oct2007
 - ROVGISUpgradeMeeting30Oct2007
- Changing ArcMapNav settings
- AUV Science Data Processing

This page last changed on Feb 17, 2022 by [kgomes](#).

Mooring + CTD

Please see the [Jira problem tracking system](#)

ROV GIS Upgrade

2007 Project

- [Original proposal](#)

Meeting presentations and notes

- 3 October 2007 "kick-off" meeting ([notes](#))

ArcMapNav

Please consult the [new documentation site](#) for information on ArcMapNav.

AUVCTD

Repo migrated to git on 22 April 2020

https://bitbucket.org/mbari/auv_ctd/src/master/

Process Documentation

- [AUVCTD Processing Data Flow Diagram](#)
- Upload script
- SSDS Portal
- [Science Data Processing](#)
- [CVS repository for Science Data Processing code](#)

Data and Metadata

- [Science Data Processing products \(surveys\)](#) (list of directories by year)
- SSDS Metadata: use [Explorer](#) to search for mission deployment names, e.g. '2007.14'
- [Classification of survey names](#) to type of survey (Canyon Axis, Shelf, etc.)
- [Archive's Science Data Processing history](#)
- Core CTD -AUVCTD tasks tracked in [jira](#)

Meeting presentations and notes

- 8 June 2007 meeting ([presentation](#) and [notes](#))

ROV Navigation

General ROV Navigation Information

- [Data Flow Diagram](#)
- [Contacts](#)
- [Current Status of Navigation Editing](#)
- [Known Navigation Issues](#)
- [Navigation Editing Procedures](#)
- [Misc. Notes](#)
- [History of Navigation Editing at MBARI](#)
- Re\edited and to\be\reviewed navigation files*

Statistics comparing Original and Edited ROV Navigation data ([notes on the stats report file format](#))

Statistics run on 17 July 2007 (McCann)

- [Spreadsheet for 2006 & 2007](#)

Statistics run on 1 August 2007 (Tab separated .txt files that may be imported into Excel)

Ventana	Tiburon
navStats1989vnta.txt	
navStats1990vnta.txt	
navStats1991vnta.txt	
navStats1992vnta.txt	
navStats1993vnta.txt	
navStats1994vnta.txt	
navStats1995vnta.txt	
navStats1996vnta.txt	
navStats1997vnta.txt	navStats1997tibr.txt
navStats1998vnta.txt	navStats1998tibr.txt
navStats1999vnta.txt	navStats1999tibr.txt
navStats2000vnta.txt	navStats2000tibr.txt
navStats2001vnta.txt	navStats2001tibr.txt
navStats2002vnta.txt	navStats2002tibr.txt
navStats2003vnta.txt	navStats2003tibr.txt
navStats2004vnta.txt	navStats2004tibr.txt
navStats2005vnta.txt	navStats2005tibr.txt
navStats2006vnta.txt	navStats2006tibr.txt

navStats2007vnta.txt	navStats2007tibr.txt
--------------------------------------	--------------------------------------

ArcMapNav Demonstration

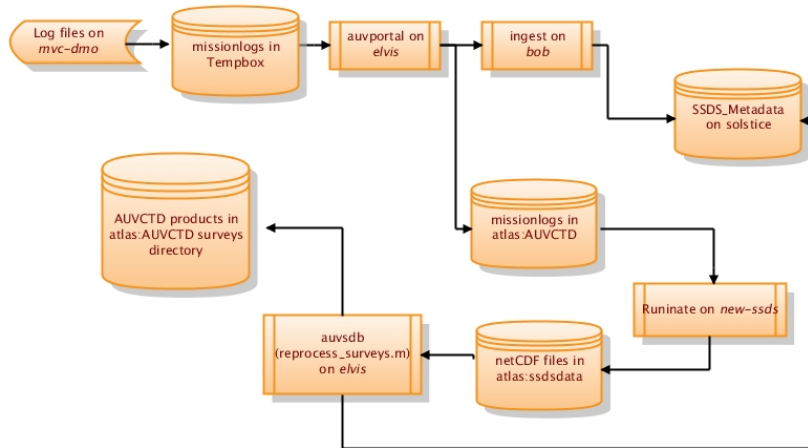
This page last changed on Feb 17, 2022 by [kgomes](#).

This documentation has moved to [this new documentation site](#)

AUVCTD Processing Data Flow Diagram

This page last changed on Feb 04, 2009 by [mccann](#).

AUVCTD Processing Data Flow Digram



Contacts

This page last changed on Jan 16, 2019 by [linda](#).

For questions regarding navigation editing contact:

Krystal Anderson (x 1962) krystle@mbari.org

DataAccessMeeting25June2008

This page last changed on Jun 30, 2008 by [godin](#).

Agenda 25 June 2008 meeting - Surface Commons

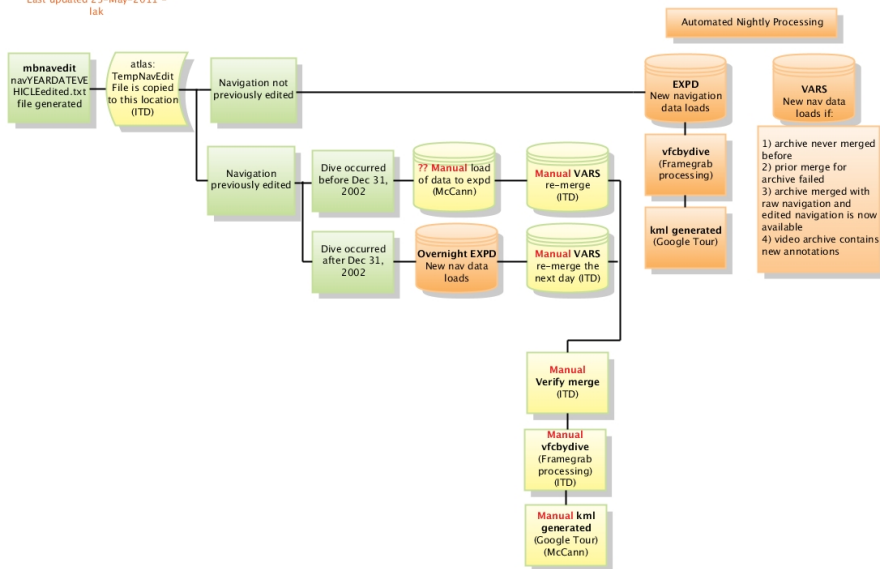
Attendees: Fred Bahr, Mark Brown, Francisco Chavez, Mike Godin, Kevin Gomes, Mike McCann, Reiko Michisaki, Brian Schlinning

1. Project goals (Francisco)
 - a. From science perspective would like to see one-stop-shopping for MBARI upper water column data.
 - b. Make it easy to create products from all of the data to answer simple science questions.
 - c. One page for mooring data that explains all the different data sets (post recovery, near real-time, etc.)
 - d. Should be able to quickly assemble data to answer basic questions.
2. Scope of work
 - a. CTD (including bottle) data - BOG
 - b. Mooring
 - c. AUVCTD
 - d. ROV-CTD
 - e. Glider line in Monterey Bay (from Scripps)
 - f. (NOEPP)
 - g. Build infrastructure in 2009 to put us in a good position for 2010.
3. Strategy to get the work done (Francisco & Mike McCann)
Regular proposal or part of the MDUC Core Data proposal?
Should we write an abstract for the July 2 deadline? (Yes. Mike McCann will write.)
Target for being an MDUC core data project.
4. Gross estimate of Support Engineering time to be requested (WAG)
 - a. Mark says that about 130 days fully allocates one support engineer
 - i. Mike McCann will work on nuts and bolts
 - ii. Brian has offered to develop toolkits (Matlab) for working with the data
 - b. We can count on about 60 days of Reiko's time
 - c. We'll request some of Kevin's time to help when we need help with SSDS
 - d. Mike Godin's time - to serve as a customer
 - e. Fred and CenCOOS are some of the customers
5. Brainstorm technical approaches

DataFlowPage

This page last changed on Dec 17, 2012 by [linda](#).

Last updated 25-May-2011 -
lak



Editing Instructions

This page last changed on Mar 03, 2009 by [linda](#).

To change read-write permissions for files created by nav editors:

- Open unix window (x-term)
- Log in
- type: `vi ~/.login` (this accesses your login file)
- down arrow to the umask line and change it from 022 to 002 (to change a character, place the cursor over the character and type `r`, then enter the new value)
- hit escape (Esc)
- hit Ctrl Z, Ctrl Z (this saves the file)
- Log off and back in OR type: `source ~/.login`

History of Navigation Editing

This page last changed on Dec 17, 2012 by [linda](#).

Navigation editing initially done by IAG beginning in 2001: Mike McCann, Rich Schramm, Brian Schlining, John Graybeal, Kevin Gomes, Andrew Chase

Navigation editing by ITD beginning sometime in 2006: Jennifer Chase, Jane Curtner, Connor

September 2007, Linda Kuhnz, Lori Chaney, Dana Lacono

January 2012, Linda Kuhnz

Historical summary of edited dives (note: we will review 1997-1999 dives in 2009 to see if anything further can be done with them):

- No dives prior to 1996 were edited
- 1997 Ventana; majority are un-edited
- 1998 Tiburon: only 3 of 30 were edited
- 1998 Ventana: majority are un-edited
- 1999 Tiburon: no editing has been done to the 15 files
- We are evaluating if there is any utility in processing these old files. Contact Linda Kuhnz if you have input.

Known Navigation Editing Issues

This page last changed on Jul 06, 2015 by [linda](#).

Recent notes and issues:

About June 2, 2015: Nav upgrade for Ventana is causing a ~150 m error.

August 30, 2013 Ventana (From Dawe); CASIUS offsets seem more reasonable than last calibration however Sonardyne is still concerned about the 'movement' of the GDT head. Also discovered there are 5 variables to enter after a CASIUS calibration. Windows XP entered these value automatically however Windows 7 requires them to be input manually.

February 20, 2013 from DJ Osborne: Just a few minor tweeks to let you know about regarding nav on the Carson. Based on Winfrogs strong suggestion we have added a sound velocity profile in sonardyne. You can turn it off in the environmental page if you have issues or otherwise. BTW, kalman filtering is off on both sonardyne and winfrog. Sonardyne is now available for viewing at the aft steering station on the bridge, left monitor, aux input. You may need to ask the pilots to activate it (alt C) No need for the sonardyne power supply to drive that head all the time when you don't need it on. A third slave for winfrog was added in the dry lab to move towards giving the AUV guys a better set up and to keep them from having to VNC or take trips into the control room. Sonardyne has given the ok for us to put in the offsets we acquired when we did the CASIUS last year. I didn't get a great deal of feedback as to the concerns of our dataset but the results are: Pitch .65, roll 2.09 and heading -.90. The calibration offsets HAVE been entered into Sonardyne from now forward.

Files for Tiburon 2004249 and 250 have been specially altered by Mike McCann using markpoint files.

March 2011: Dave Caress has potentially identified an issue with the software and is working on a new release of MBnaveedit. We are still getting un-removable spikes in the navigation files.

March 2011 Western Flyer returned from the shipyard with a new Sonardyne hydrophone; a Cassius calibration occurred on March 4, 2011 (see attached document). Navigation data from the first cruise looks much improved - lak.

April 26, 2010 Doc Ricketts is preparing to change from fusion to ranger software due to support issues.

Flat-line navigation for DOCR 2009299 and 2009309

October 2009: Dave Caress has potentially identified an issue with the software and is working on a new release of MBnaveedit. We are still getting un-removable spikes in the navigation files.

Poor navigation data coming from Doc Ricketts (specifically through D0041 on June 19, 2009).

Ventana dives V3443 - 3448 (2009303-2009310) have no usable navigation data. ROV pilots/R. Schramm reduced the NEMA message stream from Winfrog due to sonar interference. The problem was identified during V3448 and corrected about half way through the dive.

Regarding Doc Ricketts (from Rich Schramm 5/8/09): The first few dives had bad pressure from the vehicle. For sure the first 3 dives had major issues and ctd could (should?) be substituted. Then on dives less than about D0014 the parosci had issues if vehicle went past 3267.0 - which it didn't for most of the dives (it did on Smith dives- lak). I don't think SMD has sent a fix for that problem yet but we have a workaround where we intercept the sensor data and fix it before it gets to the vehicle.

April 2009: Dave Caress has released a new version of MBnaveedit to a fix for the following problem: based on the nature of specific dives, some navigation spikes cannot be removed. See V3334, V3335 as examples. We are currently evaluating the new software.

Poor navigation data coming from Doc Ricketts (through D0041 on June 19, 2009).

Historic issues:

1) Between **26 June 2008 and 03 February 2009**, there is a roundoff error logged in all raw navigation data files essentially rendering the last 2 digits of latitude and longitude meaningless. The truncation of precision was because of error in the linux port of the datamanager code. The error was identified and fixed on 3 Feb 2009. An analysis of Steve Rock's Ventana dive V3241 (a video mosaic survey) by using data from the winfroglogr files (rov nav input to datamanager) and properly edited data (from the low

precision archived nav) indicates that very little improvement in the final edited product can be had by replacing the low precision nav with nav reprocessed from the winfroglor data.

2) In 2001, there was a navigation "freezing" problem. This problem occurred from **March 25th, 2001 11:59:58 PM through December 14th, 2001 05:10:23 AM**. See John Graybeal's .ppt presentation about this issue (Sept 2001) [NavProblemsReview_Graybeal.ppt](#)

3) There was an issue with dives **T119 to T244** whereby ships navigation was logging as ROV navigation; All ROV navigation for these dives is therefore erroneous. See attached excel sheet [CompareWflyTibr-Graybeal.xls](#).

4) Specific to issue #3: As of 12-February 2009, Dave Clague and Jenny Paduan are working with Mike McCann to use the Markpoint files created during dives **T119-124, T139-147, T185-201, and T207-208** to produce smoothed navigation files to merge with the video annotations. This series of dives did not have an ROV nav file recorded since ship nav was incorrectly being placed in the ROV nav file. They are going to try to use Markpoint files to reconstruct the ROV path. See attached [map](#) for a visual image of the problem and solution.

They are also looking at the navigation of dive **T425** with Mike and trying to adjust the track so that it better fits the bathymetry. At present, although the dive plots as starting and ending at the fish trap (located during navigated dive **T427**), it is not plotting so that depths match with the underlying bathymetry. This is the dive that did not have any USBL navigation because the transponder on the vehicle was not operational. See attached [map](#) for a visual image of the problem and solution.

5) Prior to September 2005, VARS (and earlier VIMS) database queries returned navigation data from the unedited navigation files.

6) **For other problematic individual dives, see [this page](#).**

7) There were depth sensor problems between 10/22/12 (yday 296) and 11/29/12. New raw navigation files were generated, but it was not necessary to re-edit the navigation files.

Current Status of navigation Editing

This page last changed on Feb 25, 2019 by [linda](#).

Status of editing:

- As of December, 2012, editing is going well for both vehicles. Still having some issues with unexpected spikes; Dave Caress is working on a solution (new Gaussian mean version of the software began testing in April, 2009)
- New dives are edited 1 - 14 days after the completion of a cruise (excluding [known issues](#))

Misc. Notes

This page last changed on Mar 09, 2009 by [linda](#).

- Dives can be viewed in 3D replay the morning after editing or re-editing, unless the dive occurred more than 30 days ago. To view newly-edited dives older than a month old, contact Mike McCann and he will reload them.
- The nightly processing for expedition database looks back from 2004 to the present for any newly edited nav files. Edited or re-edited files prior to 2004 need to be loaded manually (contact Mike McCann).
- The nightly process for VARS does NOT recognize changes in edited navigation. Re-edited files must be reported to Lonny Lundsten (via JIRA) for re-loading to VARS.
- To change read-write permissions for files created by nav editors:
 1. Open unix window (x-term)
 2. Log in
 3. type: `vi ~/.login` (this accesses your login file)
 4. down arrow to the umask line and change it from 022 to 002 (to change a character, place the cursor over the character and type r, then enter the new value)
 5. hit escape (Esc)
 6. hit Ctrl Z, Ctrl Z (this saves the file)
 7. Log off and back in OR type: `source ~/.login`

Navigation Editing Instructions

This page last changed on Oct 30, 2019 by [krystle](#).

Copying raw navigation files to a temporary directory and opening the MBnavedit program

***NOTE:** items in red are for example purposes. You must use the appropriate year, dive number, and ROV (vnta, tibr, docr, minirov) when naming and navigating in the directories.

Nav files are stored on atlas/RovNavEdit in folders by years.

1. Currently using XQuartz for connecting to systems for nav file editing.
2. Mount a temporary directory (ex: smb://atlas.shore.mbari.org/yourshare). Files must be copied to a temporary directory before working on them to avoid overwriting original files; you will delete these copies when you are done with a session. Also mount smb://atlas.shore.mbari.org/RovNavEdit.
3. XQuartz: type in the following commands:
 - a. cd /volumes/yourshare
 - b. cp /volumes/ROVNavEdit/2016/vnta/nav2016326vnta.txt . (this copies the file to the temp location for editing). For some reason, you will get an error saying that permission is denied, but note that the file IS copied and all is well. MiniROV files take the form of NAV_090517000000.mb165, but open the same as the .txt files.
 - c. ssh -Y menard (Sign in using your password; you may to ask for permissions (Dave Caress) the first time. Wait until you get the new prompt -bash-4.2\$ (this currently takes a minute)
 - d. mbnavedit -X -N

Configuring the MBnavedit program window

When the mbnavedit program opens, you need to configure it properly.

1. Uncheck the boxes for "time interval plot" and "heading plot"
2. Uncheck all the "show original data" check boxes
3. Click the CONTROLS button and choose "Nav Modeling" from the pull down menu
4. Click on "Gaussian Mean" or "inversion"
 - a. When using Gaussian mean, use the slider titled Gaussian Mean Time Window to adjust the mean value. ~47 is a good starting point. When using Inversion, set the Speed Deviation and Acceleration fields to 5. You can adjust how tight the blue line fits between raw data points by adjusting the mean value with the slider. The lower the value the tighter the line and vice versa.
 - b. Click "Apply"

Move the Nav Model panel somewhere on your screen where it is not completely overlapping the main MBnavedit window

Click the "File" button in the upper left corner of the main MBnavedit window. Choose the file you want to edit, then click O.K.

If you get an error dialog box that says "Duplicate or reverse order time stamps", Click "Control" and select Time interpolation and click 'Apply'.

Editing a dive

Start with **Sonar Depth** panel at bottom.

Use "Pick" or "Select" radio button to identify depth points outside of reasonable range. When outlying points are selected, click "Interpolate" button.
Now zoom in to look at a smaller region of the track.

Click "Pick Zoom" button, then left click (for left boundary) and middle click (for right boundary) of area to zoom in....then right click in the middle of the area.

Right mouse button will move the graph to the right, middle button will move to the left. Pick out outlying point and interpolate those.

Click "Show All" button to zoom out and see the whole track again.

Now, move up to the **Latitude** and **Longitude** panels.

The Navigation Model was applied when the file opened, so the blue line shows the estimated navigation path.

Use "Pick Zoom" to zoom in on the beginning edge case. Select points in the pre-dive area and a few after the dive began, "Flag" and "Apply" the navigation model. Move right along the data page, select outlying points, then "Flag", and "Apply" within each window as you go. By applying at each window, you can see the effect of your actions; if something didn't work out as expected, you can fix it now. The goal is to have no spikes in the Speed plot over 10 km/hour. If you do, zoom in to these spikes on the Latitude and Longitude plots and see if you can make further corrections.

To Finish the dive, first click 'Show All' then click "Select All", click in the longitude window and then "Use Solution"

* If two dives occurred on one day (this typically happens with Tiburon and Doc Ricketts), you will see two distinct entities in the sonar depth plot. Work on each dive individually by selecting, and flagging points as described above. Once both dives have been smoothed, "Select All" dives and "Use Solution" .

Copying the edited file back to the atlas/RovNavEdit directory

After quitting MBnavedit go back to the terminal window and type in the following command:

```
cp (space)nav2009221vntaedited.txt (space)/mbari/RovNavEdit/2009/vnta
```

** Once you've completed editing files, delete the files from the tmp folder you used

If there are severe navigation scatter problems, or long drop-outs of data on a particular dives (or set of dives), document this in the Expedition Database. Go to the cruise and "Edit Postcruise". In the data comments box, put in a note (ex: Navigation for this dive is scattered and somewhat unreliable, Drop off of navigation data between 20:25 and 20:34, etc.

Additional hints and tips for Navediting

This page last changed on Mar 02, 2009 by [linda](#).

You can view dives within the last 30 days with 3D replay. To view dives older than a month, contact Mike McCann and he will reload them for viewing

Expedition database looks for the modified date on re-edited dives; edited and re-edited dives should load automatically overnight.

Navigation to be reviewed for re-edit

This page last changed on Mar 04, 2009 by [linda](#).

The following files are being reviewed so see if they can be re-edited.

nav2007157vnta.txt
nav2007292vnta.txt
nav2007263tibr.txt
nav2006257vnta.txt
nav2006089tibr.txt
nav2006180tibr.txt
nav2006224tibr.txt
nav2006237tibr.txt
nav2005284vnta.txt
nav2005285vnta.txt
nav2005336vnta.txt
nav2005035tibr.txt
nav2005230tibr.txt
nav2005335tibr.txt
nav2004022tibr.txt
nav2004054tibr.txt
nav2004246tibr.txt
nav2004247tibr.txt
nav2004250tibr.txt
nav2004251tibr.txt
nav2004252tibr.txt
nav2004273tibr.txt
nav2004278tibr.txt
nav2004279tibr.txt
nav2004280tibr.txt
nav2004281tibr.txt
nav2004282tibr.txt
nav2004283tibr.txt
nav2004284tibr.txt

nav2004307tibr.txt
nav2004253vnta.txt
nav2004323vnta.txt
nav2004336vnta.txt
nav2003280tibr.txt
nav2003282tibr.txt
nav2003349vnta.txt
nav2002202tibr.txt
nav2002206tibr.txt
nav1999007vnta.txt
nav1999008vnta.txt
nav1999019vnta.txt
nav1999021vnta.txt
nav1999028vnta.txt

navStatsNotes

This page last changed on Aug 01, 2007 by [mccann](#).

The columns in the files are:

File	file name
esecs	Start Epoch Seconds (seconds after 1/1/1970 00:00 GMT)
lat	Average Latitude in the file
lon	Average Longitude in the file
maxPres	Maximum pressure in the file
numFreezes	Number of times the system logged the same exact positions (Used to analyze problems the Tiburon USBL system in 2002)
totalDuration	Total duration in minutes of the freezes (Used to analyze problems the Tiburon USBL system in 2002)
recordDuration	Count of records (converted to minutes) when the position was frozen (Used to analyze problems the Tiburon USBL system in 2002)
aveSD	The aveSD column is the average standard deviation between the original and edited nav. If the values are 0.00 then the original and edited data are identical. Larger values will indicate a degraded performance in the nav system - or an incorrectly edited file. There will naturally be a correlation with performance accuracy and depth.
editedNav	Whether an Edited nav file exists or not

Here is the Perl Script that produces the Statistics reports:

```
#!/usr/bin/perl
#
# Script to print frozen position statistics for Ventana nav data
#
# Mike McCann 25 August 2001 MBARI
#
# $Id: navStats.pl,v 1.3 2007/06/06 21:39:16 mccann Exp mccann $
#
# $Log: navStats.pl,v $
# Revision 1.3 2007/06/06 21:39:16 mccann
# Made work on elvis. Added comment.
#
# Revision 1.2 2002/01/18 17:36:51 mccann
```

```

# Added debug flag and proper checks for tibr data files.
#
# Revision 1.1 2002/01/18 17:34:33 mccann
# Initial revision
#

use Statistics::Descriptive;

my $debug = 0;

die "\nUsage: $0 <directory where *vnta.txt files are> [#frozen Pos] [excel]\n\n" unless -d
$ARGV[0];

$showOver = $ARGV[1] || 60;      # Show if pos repeats for more times than this
$outputExcel = 0;
$outputExcel = 1 if ( $ARGV[2] =~ /excel/i );
$| = 1;
$deltaT = 5 / 60;

#
# Open directory and get list of all the raw nav files in it. Handle vnta & tibr
#
opendir (NAVDIR, $ARGV[0]) || die "Can't open directory $ARGV[0]: $!\n";
if ( $ARGV[0] =~ /vnta/ ) {
@navfiles = grep { /vnta.txt$/ } readdir (NAVDIR);
}
elsif ( $ARGV[0] =~ /tibr/ ) {
@navfiles = grep { /tibr.txt$/ } readdir (NAVDIR);
}
else {
die "$0 works only for vnta or tibr directories.\n";
}
die "Empty \@navFiles list. Was reading from: $ARGV[0]\n" unless @navfiles;
closedir (NAVDIR);

#
# Print header for output
# - the aveSD column is the average standard deviation between the original and edited nav
#   if the values are 0.00 then the original and edited data are identical.  Larger
#   values will indicate a degraded performance in the nav system.  There will natuarlly
#   be a correlation with performance accuracy and depth.
#
if ( $outputExcel ) {
##print "File,esecs,Lat,Lon,maxPres,numFreezes,totalDuration,recordDuration,aveSD,editedData\n";
print "File\tesecs\tLat\tLon\tmaxPres\tnumFreezes\ttotalDuration\trecordDuration\taveSD\teditedData
\n";
}
else {
print "File          str (hhmmss) end   froz Lat      froz lon   sPres    ePres    #    duration\n";
}

#
# Loop through all the files and collect stats on frozen positions
#
foreach ( @navfiles ) {
$file = $_;
print "file = $file\n" if $debug;
if ( $outputExcel ) {
}
else {
print "$file\n";
}

calcFreezes();

calcSD();

```

```

} # End foreach

#-----
# Subroutines: just to partition the work.  No args passed.  Use global vars
#

sub calcFreezes {

open (NAVFILE, "$ARGV[0]/${file}") || die "Can't open file $ARGV[0]/${file}: $!\n";
$oldLat = 0;
$oldLon = 0;
$sameCount = 0;
$notSame = 0;
$prFile = $file;          # Print file string, for listing below
$prFile =~ s/vnta.txt//;
$timeSum = 0;
$recSum = 0;
$numFrozen = 0;
$maxPres = 0;
while ( <NAVFILE> ) {
next unless /^\\d/;        # Skip rec unless it begins with number

#
# Read record into my variables
#
($Year, $Day, $Time, $Usec, $Lat, $Lon, $East, $North,
$Pres, $Head, $Alti, $Pitch, $Roll, $PosFlag, $PresFlag,
$HeadFlag, $AltiFlag, $AttitFlag) = split (' ', $);
if ( $maxPres < $Pres ) {
$maxPres = $Pres;
}

#
# Count records of unchanging Lat & Lon
#
print "$_\\n" if $debug;
if ( $Lat == $oldLat && $Lon == $oldLon ) {
$sameCount++;
if ( $sameCount == 1 ) {
$begUsec = $Usec;
$begTime = $Time;
$begPres = $Pres;
}
}
elseif ( $oldLat != 0 ) {
$notSame = 1;
$timeDiffSec = $Usec - $begUsec;
$duration = sprintf("%6.1f min", $timeDiffSec / 60);
} # End if ( $Lat ...

#
# Print out stats for frozen position after we've counted them
#
if ( $sameCount > $showOver && $notSame ) {
if ( $outputExcel ) {
}
else {
printf ("%10s %d to %d %f %f %5.1f to %5.1f %3d %s\\n",
$prFile, $begTime, $Time, $oldLat, $oldLon, $begPres,
$Pres, $sameCount, $duration);
$prFile = ' ';
}
}
$recSum += $deltaT * $sameCount;

```

```

$sameCount = 0;
$notSame = 0;
$timeSum += $duration;
$numFrozen++;
##die "Debugging\n";
}

#
# Reset counter if not over our $showOver criteria
#
elseif ( $sameCount <= $showOver && $notSame ) {
$sameCount = 0;
$notSame = 0;
} # End if ( $sameCount >

$oldLat = $Lat;
$oldLon = $Lon;

} # End while (NAVFILE

#
# Print summary for this day
#

if ( $outputExcel ) {
##print "$prFile,$Usec,$Lat,$Lon,$maxPres,$numFrozen,$timeSum,", sprintf("%.6f", $recSum);
print "$prFile\t$Usec\t$Lat\t$Lon\t$maxPres\t$numFrozen\t$timeSum\t", sprintf("%.6f", $recSum);
}
else {
if ( $numFrozen > 0 ) {
printf ( "          %3d frozen positions over %3d records long lasting total of %5.1f minutes\n",
$numFrozen, $showOver, $timeSum);
}
} # End if ( $outputExcel

} # End calcFreezes()

sub calcSD {

#
# Create list of lat & lon of differences for the day
#
open (NAVFILE, "$ARGV[0]/${file}") || die "Can't open file $ARGV[0]/${file}: $!\n";

$editedFile = $file;
$editedFile =~ s/\.txt/edited.txt/;

if ( -f "$ARGV[0]/${editedFile}" ) {
open (EDITEDNAV, "$ARGV[0]/${editedFile}") ||
die "Can't open file $ARGV[0]/${editedFile}: $!\n";
}
else {
print STDERR "Can't open file $ARGV[0]/${editedFile}: $!\n";
print STDERR "Skipping this file.\n";
if ( $outputExcel ) {
##print ",0,NoEditedNav\n";
print "\t0\tNoEditedNav\n";
}
return;
}

undef @lonDiff;
undef @latDiff;

while ( ($raw = <NAVFILE>) && ($edited = <EDITEDNAV>) ) {

```

```

next unless $raw =~ /\^d/;      # Skip rec unless it begins with number

#
# Read raw record into my variables
#
my ($Year, $Day, $Time, $Usec, $Lat, $Lon, $East, $North,
$Pres, $Head, $Alti, $Pitch, $Roll, $PosFlag, $PresFlag,
$HeadFlag, $AltiFlag, $AttitFlag) = split ('', $raw);

#
# Read edited record into my variables
#
my (undef, undef, undef, $eUsec, $eLat, $eLon, undef, undef,
$ePres, undef, undef, undef, undef, undef, undef,
undef, undef, undef) = split ('', $edited);

##print "Lat diff = ", $Lat - $eLat, "  Lon diff = ", $Lon - $eLon, "\n";

push @latDiff, $Lat - $eLat;
push @lonDiff, $Lon - $eLon;

} # End while ( <NAVFILE>

#
# Compute stats on pos differences
#
undef $lonStats;
undef $latStats;
my $lonStats = Statistics::Descriptive::Full->new();
my $latStats = Statistics::Descriptive::Full->new();
$lonStats->add_data(@lonDiff);
$latStats->add_data(@latDiff);

if ( $outputExcel ) {
printf "\t %9.6f\n", (
sqrt($lonStats->variance) + sqrt($latStats->variance) ) / 2;
}
else {
printf "      lonDiff SD = %9.6f  latDiff SD = %9.6f,  Ave = %9.6f\n",
sqrt($lonStats->variance), sqrt($latStats->variance),
(sqrt($lonStats->variance) + sqrt($latStats->variance) ) / 2;
}

} # End calcSD()

```

The Perl script is executed with this shell script:

```

#!/bin/sh
#
# Run stats on all of our ROV nav data

outDir="/hosts/tempest/tempbox/mccann/navstats"

for rovr in tibr vnta
do
for yrdir in /hosts/tornado/TempNav/200[6-7]
do
dir="$yrdir/$rovr"
year=`echo $yrdir | cut -d/ -f5`
echo navStats.pl $dir 60 excel ">" $outDir/navStats${year}${rovr}.txt
/bin/rm -f $outDir/navStats${year}${rovr}.txt
navStats.pl $dir 60 excel > $outDir/navStats${year}${rovr}.txt
done
#
# Concatenate all the ROV stat files into one

```

```
#  
cat $outDir/navStats????${rov}.txt > $outDir/navStats${rov}.txt  
done
```

PostgreSQL Server kraken setup

This page last changed on Jun 27, 2013 by [mccann](#).

PostgreSQL Server kraken setup

This page is for recording significant configuration settings for the dedicated PostgreSQL server kraken. It exists in concert with the [wiki page for the pgtest server](#) where various system and PostgreSQL parameters are tuned for optimal performance on a ESX virtual machine. The parameters for kraken will be different than for pgtest; however, pgtest provides a useful test environment for proposed changes.

Hardware Documentation

- [FusionIO "disk drive" Release Notes](#)
- [FusionIO "disk drive" User Guide for Linux](#)

Comments

```
sudo yum install gnuplot
```

Posted by [mccann](#) at Jul 10, 2013.

.....

Configuring kraken to run pgbench-tools as user mccann on kraken:

```
cd /var/www/html
sudo mkdir pgbench-tools
sudo chown postgres pgbench-tools
```

Copy pgbench-tools from its special configuration on pgtest for passwordless connection to the database, as postgres@pgtest:

```
cd /var/lib/pgsql/dev/pgbench-tools
tar cvf /tmp/pgbench-tools.tar *
```

Moved the .tar file from pgtest to kraken, then install in the web directory rather than in postgres's home directory. As postgres on kraken:

```
cd /var/www/html/pgbench-tools
cp /tmp/pgbench-tools.tar .
tar xvf pgbench-tools.tar
```

Setup pgbench-tools databases and run an initial test as postgres@kraken:

```
createdb results
createdb pgbench
cd /var/www/html/pgbench-tools
psql -f init/resultdb.sql -d results
sudo yum install postgresql91-contrib # as mccann (needed for the pgbench binary)
./runset
```

Configure web server on kraken to serve up results from pgbench-tools. First, submit help ticket to open up port 80 for http on kraken.

```
sudo chkconfig httpd on
sudo /sbin/service httpd start
```

Posted by [mccann](#) at Jul 11, 2013.

.....

Modifying system parameters that have been tested on pgtest. Here are the changes:

```
<197 kraken.shore.mbari.org //etc> diff sysctl.conf.130712 sysctl.conf
40a41,53
>
> # Swappiness set to 0 per p.81 in PostgreSQL 9.0 High Performance book. mpm - 12 July 2013
> vm.swappiness = 0
```

```
> # Reduce tendency to allocate more ram than the system has. mpm - 12 July 2013
> vm.overcommit_memory = 2
>
> # Setting for shared memory = 1/2 physical RAM = 64 GB. mpm 12 July 2013
> ##kernel.shmmax = 67649847296
> ##kernel.shmall = 16516076
>
> # Set shared memory to 4 GB - a better value for PostgreSQL on large memory systems. mpm 12 July 2013
> kernel.shmmax = 4228112384
> kernel.shmall = 1032254
```

```
<198 kraken.shore.mbari.org //etc> diff fstab.130712 fstab
11,13c11,13
< /dev/mapper/vg_kraken-LogVol02 /pg_xlog_spinning          ext4      defaults          1 2
< /dev/mapper/vg_kraken-LogVol04 /pgdata_spinning          ext4      defaults          1 2
< /dev/mapper/vg_kraken-LogVol03 /tablespace_spinning       ext4      defaults          1 2
---
> /dev/mapper/vg_kraken-LogVol02 /pg_xlog_spinning          ext4      defaults,noatime    1 2
> /dev/mapper/vg_kraken-LogVol04 /pgdata_spinning          ext4      defaults,noatime    1 2
> /dev/mapper/vg_kraken-LogVol03 /tablespace_spinning       ext4      defaults,noatime    1 2
22,24c22,24
< /dev/mapper/iomemory_vg-iomemVol1 /pg_xlog              ext4      defaults,noauto,discard 0 0
< /dev/mapper/iomemory_vg-iomemVol2 /tablespace           ext4      defaults,noauto,discard 0 0
< /dev/mapper/iomemory_vg-iomemVol3 /pgdata              ext4      defaults,noauto,discard 0 0
---
> /dev/mapper/iomemory_vg-iomemVol1 /pg_xlog              ext4      defaults,noatime,noauto,discard 0 0
> /dev/mapper/iomemory_vg-iomemVol2 /tablespace           ext4      defaults,noatime,noauto,discard 0 0
> /dev/mapper/iomemory_vg-iomemVol3 /pgdata              ext4      defaults,noatime,noauto,discard 0 0
```

```
<199 kraken.shore.mbari.org //etc> diff rc.local.130712 rc.local
7a8,9
> blockdev --setra 8192 /dev/mapper/vg_kraken-LogVol04
> blockdev --setra 8192 /dev/mapper/iomemory_vg-iomemVol3
```

pgbench-tools results at: http://kraken.shore.mbari.org/pgbench-tools/results.130715.AfterOSConfigChanges_tests_61_132/index.htm

Posted by [mccann](#) at Jul 12, 2013.

Modified postgresql.conf based on lessons learned on pgtest. Here's the diff:

```
-bash-4.1$ diff postgresql.conf.130716 postgresql.conf
59c59,60
< #listen_addresses = 'localhost'          # what IP address(es) to listen on;
---
> # Allow connections from mbari hosts inside our firewall - mpm 7 May 2013
> listen_addresses = '*'                  # what IP address(es) to listen on;
64c65,66
< max_connections = 100                    # (change requires restart)
---
> # Changed max_connections from 100 to 10 corresponding to increase in work_mem - mpm 22 May 2013
> max_connections = 10                    # (change requires restart)
109c111
< shared_buffers = 32MB                    # min 128kB
---
> shared_buffers = 3GB                    # min 128kB
118c120,122
< #work_mem = 1MB                          # min 64kB
---
>
> # Changed from 1 MB to 400MB to support large STQs sorts - mpm 22 May 2013
> work_mem = 400MB                        # min 64kB
174c178,179
< #checkpoint_segments = 3                  # in logfile segments, min 1, 16MB each
---
```

```

> # pgbench-tools indicated high checkpoint frequency, adjust with guidance from p.137. dr 3 July 2013
> checkpoint_segments = 10          # in logfile segments, min 1, 16MB each
176c181
< #checkpoint_completion_target = 0.5 # checkpoint target duration, 0.0 - 1.0
---
> checkpoint_completion_target = 0.9 # checkpoint target duration, 0.0 - 1.0
244c249,251
< #random_page_cost = 4.0          # same scale as above
---
>
> # Changed to 1.01 per Greg Smith's suggestion on postgresql-perform list - seems to improve STOQS
mp query performance - mpm 14 May 2013
> random_page_cost = 1.01          # same scale as above
359c366,367
< #log_min_duration_statement = -1  # -1 is disabled, 0 logs all statements
---
> # Log all queries taking longer than 1 second. mpm 15 July 2013
> log_min_duration_statement = 1000 # -1 is disabled, 0 logs all statements
377c385,387
< #log_line_prefix = ''            # special values:
---
> # Set timestamp for log files to be compatible with pgFouine. mpm 15 July 2013
> log_line_prefix = '%t [%p]: [%l-1] user=%u, db=%d'
>                                # special values:
399c409,410
< #log_temp_files = -1            # log temporary files equal or larger
---
> # Turned on logging of temp files. See $PGDATA/base/<db oid>/pgsql_tmp for what a DB is doing -
mpm 15 May 2013
> log_temp_files = 0              # log temporary files equal or larger
482c493,494
< #timezone = '(defaults to server environment setting)'
---
> # Servers running STOQS databases must have timezone set to 'GMT'
> timezone = 'GMT'

```

pgbench-tools results at: http://kraken.shore.mbari.org/pgbench-tools/results.130716.AfterPostgresql.confChanges_tests_133_204/index.htm (as expected, no appreciable changes in performance)

Posted by [mccann](#) at Jul 16, 2013.

Installed STOQS on kraken in /u/stoqsadm/dev/stoqshg_kraken. First time installing on a system with limited sudo privileges. The 'make install's for gdal and mapserver required help tickets for real root to do. Several additional packages needed to be installed and some problems were discovered when running loaders/loadTestData.py. Details documented in these change sets:

<https://code.google.com/p/stoqs/source/detail?r=fa16cdaf4acd4e7d4dd793c9eee9ca9319d21ff0>
<https://code.google.com/p/stoqs/source/detail?r=be90b1e585156e3816e488fe3f10f49f2852ace9>

Posted by [mccann](#) at Jul 17, 2013.

Fixed problems with stoqs install on kraken:

1. Mapserver needed to be rebuilt with the configure argument: --with-postgis=/usr/pgsql-9.1/bin/pg_config and then copied to /var/www/cgi-bin
2. Was getting "undefined symbol: GDALSetRasterUnitType" when trying to load web page. This was caused by Django not being able to find the locally built gdal. Instructions to add "export LD_LIBRARY_PATH='/usr/local/lib'" to the httpd startup script was added to the PRODUCTION file.
3. We also needed to add this to the httpd startup script: export GDAL_DATA='/usr/share/gdal/'.

Posted by [mccann](#) at Jul 19, 2013.

Doing a little exercise of STOQS on kraken. So far performance is similar to odss-staging. Turned down log_min_duration to 100ms in order to see some queries logged.

Installing pgFouine to analyze postgresql logs:

```
sudo yum install pgfouine
```

Edit timezone setting in /etc/php.ini:

```
> diff /etc/php.ini.130723 /etc/php.ini
946c946
< ;date.timezone =
---
> date.timezone = America/Los_Angeles
```

Create a web directory for pgFouine reports and test running it:

```
sudo mkdir /var/www/html/pgfouine
sudo chown postgres /var/www/html/pgfouine
pgfouine.php -logtype stderr -file /var/lib/pgsql/9.1/data/pg_log/postgresql-Mon.log -format html
> /var/www/html/pgfouine/InitialTest.html
```

Results viewable at <http://kraken.shore.mbari.org/pgfouine/InitialTest.html>

Posted by [mccann](#) at Jul 23, 2013.

Configured Postgresql logging for production operation with datestamped log files and decreased log_min_duration to 10ms to capture more STOQS query activity:

```
$ diff postgresql.conf.130723 postgresql.conf
297c297
< log_filename = 'postgresql-%a.log'      # log file name pattern,
---
> log_filename = 'postgresql-%Y%m%d.log'    # log file name pattern,
367c367
< log_min_duration_statement = 100        # -1 is disabled, 0 logs all statements
---
> log_min_duration_statement = 10          # -1 is disabled, 0 logs all statements
```

pgFouine now:

```
pgfouine.php -logtype stderr -file /var/lib/pgsql/9.1/data/pg_log/postgresql-20130723.log -format
html > /var/www/html/pgfouine/postgresql-20130723.html
```

gives: <http://kraken.shore.mbari.org/pgfouine/postgresql-20130723.html>

The pgfouine.php command is something we can put into cron, or just run it when we need to and examine the reports in <http://kraken.shore.mbari.org/pgfouine/>.

Posted by [mccann](#) at Jul 23, 2013.

Last Thursday afternoon we exercised kraken's FusionIO drives with multiple parallel loads of STOQS campaign data. About 6 parallel loads were started around 2pm local time. Here is the sar report for that day as seen with this command:

```
sar -f /var/log/sa/sa01 -b -d > kraken_sar_1aug2013.txt
```

[kraken_sar_1aug2013.txt](#)

The highest %util was 12.59% at 2:20 PM. We're barely scratching the surface with the capabilities of this machine.

Posted by [mccann](#) at Aug 05, 2013.

PostgreSQL-9.1 Server Tuning

This page last changed on Jul 03, 2013 by [mccann](#).

PostgreSQL-9.1 Server Tuning

Configuring a Postgres server to efficiently serve data is an involved exercise, but one that can be successful if a deliberate strategy is followed. This page exists to track setup and configuration changes and their results with the goal of making informed decisions on tuning a server for a particular application. We'll start with a STOQS database running on a VM configured according to WHD-16322:

```
Request_Submitted_By: mccann@mbari.org
VM_Name: pgtest
VM_Purpose: To test PostgreSQL tuning parameters for current and future geospatial databases
VM_Expiration: 1 year
VM_Support: NOT_IS_Supported
Alt_Administrator: Mike McCann
Project_Number:
VM_OS: CentOS 6 - Recommended for Linux
RAM: 4
Addl_RAM_Reason: Additional RAM needed to cache database requests and this needs to be closer to
the production server value of 64 GB.
VM_disk: 25 GB total space is fine, but I request 4 separate devices for mounting different "disks"
for optimal database performance. Here is the requested layout:
```

```
/(root) - Operating system [10 GB - or what you think is really needed]
$PGDATA - Database data [10 GB]
$PGDATA/pg_xlog - DB Write Ahead Log (WAL) [3 GB]
Tablespace - Temporary files [2 GB]
```

VM_Comments: I request root (or sudo to everything) on the system.

The system really is temporary. Once the production PostgreSQL server is in place and configured (by September 2013) this server will no longer be needed,

6 May 2013

After the VM was setup the following was done:

1. Installed PostgreSQL-9.1 and postgis via yum(1)
2. Initialized cluster with "/sbin/service postgresql-9.1 initdb"
3. Moved px_xlog, pg_tablespace, data to their own partitions

Current disk/directory layout on pgtest:

```
[root@pgtest 9.1]# mount -t ext4
/dev/mapper/vg_templatecentos64-lv_root on / type ext4 (rw)
/dev/sda1 on /boot type ext4 (rw)
/dev/sdb1 on /pgdata type ext4 (rw)
/dev/sdc1 on /pg_xlog type ext4 (rw)
/dev/sdd1 on /tablespace type ext4 (rw)
[root@pgtest 9.1]# ls -l
total 8
drwx----- 2 postgres postgres 4096 Apr  2 09:24 backups
lrwxrwxrwx 1 root    root        12 May  6 16:02 data -> /pgdata/data
-rw----- 1 postgres postgres 1749 May  6 16:03 pgstartup.log
[root@pgtest 9.1]# cd data
[root@pgtest data]# ls -l
total 84
drwx----- 6 postgres postgres 4096 May  6 16:05 base
drwx----- 2 postgres postgres 4096 May  6 16:12 global
drwx----- 2 postgres postgres 4096 May  6 15:29 pg_clog
-rw----- 1 postgres postgres 4232 May  6 15:29 pg_hba.conf
-rw----- 1 postgres postgres 1636 May  6 15:29 pg_ident.conf
drwx----- 2 postgres postgres 4096 May  6 15:33 pg_log
drwx----- 4 postgres postgres 4096 May  6 15:29 pg_multixact
```

```

drwx----- 2 postgres postgres 4096 May 6 16:03 pg_notify
drwx----- 2 postgres postgres 4096 May 6 15:29 pg_serial
drwx----- 2 postgres postgres 4096 May 6 16:30 pg_stat_tmp
drwx----- 2 postgres postgres 4096 May 6 15:29 pg_subtrans
lrwxrwxrwx 1 root root 21 May 6 15:57 pg_tblspc -> /tablespace/pg_tblspc
drwx----- 2 postgres postgres 4096 May 6 15:29 pg_twophase
-rw----- 1 postgres postgres 4 May 6 15:29 PG_VERSION
lrwxrwxrwx 1 root root 16 May 6 15:38 pg_xlog -> /pg_xlog/pg_xlog
-rw----- 1 postgres postgres 19146 May 6 15:29 postgresql.conf
-rw----- 1 postgres postgres 71 May 6 16:03 postmaster.opts
-rw----- 1 postgres postgres 81 May 6 16:03 postmaster.pid

```

Starting on changing default system configurations (previous .conf file moved to <file>.YYMMDD file, copied to <file> and edited):

1. /etc/sysctl.conf: vm.swappiness = 0 and vm.overcommit_memory = 2
2. /etc/fstab: Set noatime on the database partitions
3. /etc/rc.local: blockdev --setra 8192 /dev/sdb1 (pgdata)

7 May 2013

Postgresql & system firewall configuration changes to allow connections across the network (see diffs from .130507 conf files):

1. Edited /var/lib/pgsql/9.1/data/pg_hba.conf
2. Edited /var/lib/pgsql/9.1/data/postgresql.conf
3. Executed "sudo python /usr/bin/system-config-firewall" and added 5432:tcp & 5432:udp

From machine odss-test:

1. Added stoqs_september2012_pgtest to ~odssadm/dev/stoqshg/privateSettings pointing to the database on pgtest
2. Added stoqsadm user and created stoqs_september2012 database according to instructions in the STOQS INSTALL file
3. Started "~odssadm/dev/stoqshg/loaders/CANON/loadCANON_september2012.py 1 stoqs_september2012_pgtest" at 7 May 2013 1550 PDT
4. Monitoring vmstat 1 and iostat 1 on pgtest
 - a. vmstat showing no swap activity. Good.
 - b. iostat showing ~215 tps and ~3500 Blk_wrtn/s to sdc, which is /pg_xlog every second
 - c. iostat showing 64 to 500 Blk_wrtn/s every 3 seconds or so to sdb, which is /pgdata

8 May 2013

stoqs_march2013_s database copied to pgtest from odss-test by Neil using pg_dump / pg_restore to serve as a good test database that has 3,381,366 data values. Initial tests shows comparable performance between the databases accessible from the stoqs ui at http://odss-test.shore.mbari.org/canon/stoqs_march2013_s/query/ and http://odss-test.shore.mbari.org/canon/stoqs_march2013_s_pgtest/query/. Direct queries from psql on each machine are also comparable, e.g.:

On odss-test:

```
stoqs_march2013_s=# explain analyze select * from stoqs_measuredparameter order by datavalue;
QUERY PLAN
```

```

-----
Sort  (cost=629955.51..638407.16 rows=3380659 width=20) (actual time=6082.363..6971.959
rows=3381814 loops=1)
Sort Key: datavalue
Sort Method: external merge  Disk: 112424kB
-> Seq Scan on stoqs_measuredparameter  (cost=0.00..55347.59 rows=3380659 width=20) (actual
time=0.020..390.344 rows=3381814 loops=1)
Total runtime: 7133.318 ms
(5 rows)

```

On pgtest:

```
stoqs_march2013_s=# explain analyze select * from stoqs_measuredparameter order by datavalue;
QUERY PLAN
```

```

-----
Sort (cost=630174.15..638628.68 rows=3381814 width=20) (actual time=6214.177..7099.968
rows=3381814 loops=1)
Sort Key: datavalue
Sort Method: external merge Disk: 112424kB
-> Seq Scan on stoqs_measuredparameter (cost=0.00..55359.14 rows=3381814 width=20) (actual
time=0.011..398.572 rows=3381814 loops=1)
Total runtime: 7266.122 ms
(5 rows)

```

Changing shared memory from initial value of 32 MB to 2 GB (half of all physical memory)

First execute script from Page 101 in Postgres 9.0 High Performance book:

```

<154 pgtest.shore.mbari.org /u/mccann> cat ./pg_smmem.sh
#!/bin/bash
# simple shmsetup script
page_size=`getconf PAGE_SIZE`
phys_pages=`getconf _PHYS_PAGES`
shmall=`expr $phys_pages / 2`
shmmax=`expr $shmall \* $page_size`
echo kernel.shmmax = $shmmax
echo kernel.shmall = $shmall

```

```

<155 pgtest.shore.mbari.org /u/mccann> ./pg_smmem.sh
kernel.shmmax = 2009198592
kernel.shmall = 490527

```

Modified system and postgres .conf files:

1. /etc/sysctl.conf with kernel values from above
2. postgresql.conf with shared_buffers = 1 GB (I tried setting it to 2 GB, but got an error when restarting postgresql-9.1; our test db is only 778 MB anyway)

No joy for immediate increase in performance. After rebooting the "explain analyze select * from stoqs_measuredparameter order by datavalue;" query still takes about 7.5 seconds.

Installed postgres contrib package in order to get tools for peeking at Postgres's memory cache: "yum install postgresql91-contrib.x86_64". Continuing to read Chapter 5...

10 May 2013

Some work_mem setting tests:

```

stoqs_march2013_s=# show work_mem;
work_mem
-----
1MB
(1 row)

stoqs_march2013_s=# set work_mem='354MB';
SET
stoqs_march2013_s=# explain analyze select * from stoqs_measuredparameter order by datavalue;
QUERY PLAN
-----
Sort (cost=422106.15..430560.68 rows=3381814 width=20) (actual time=7276.353..7659.802
rows=3381814 loops=1)
Sort Key: datavalue
Sort Method: external sort Disk: 112400kB
-> Seq Scan on stoqs_measuredparameter (cost=0.00..55359.14 rows=3381814 width=20) (actual
time=0.048..382.085 rows=3381814 loops=1)
Total runtime: 7828.067 ms
(5 rows)

stoqs_march2013_s=# set work_mem='355MB';
SET

```

```
stoqs_march2013_s=# explain analyze select * from stoqs_measuredparameter order by datavalue;
QUERY PLAN
-----
Sort  (cost=422106.15..430560.68 rows=3381814 width=20) (actual time=2399.499..2833.767
rows=3381814 loops=1)
Sort Key: datavalue
Sort Method: quicksort  Memory: 362509kB
-> Seq Scan on stoqs_measuredparameter  (cost=0.00..55359.14 rows=3381814 width=20) (actual
time=0.011..357.892 rows=3381814 loops=1)
Total runtime: 2998.478 ms
(5 rows)
```

Bingo! Passing a threshold to 355MB of work_mem lets the sort happen as a quicksort in memory decreasing query time by better than half!

Suggestions at <http://stackoverflow.com/questions/8106181/work-mem-in-postgresql-on-linux> and http://www.depesz.com/2011/07/03/understanding-postgresql-conf-work_mem/ are that the application can temporarily set the work_mem to an appropriate value for the queries that need more than the default value and then reset it when the query is done. Setting it globally to a large value can cause unneeded use of work_mem leading to out of memory errors.

13 May 2013

Following up on 8 May 2013 note, finishing pg_buffercache install. Note: instructions on p. 105 are not correct for pgsq-9.1.

This completes the install and tests that it is installed:

```
postgres=# create extension pg_buffercache;
CREATE EXTENSION
postgres=# \dx
List of installed extensions
Name          | Version | Schema  | Description
-----+-----+-----+-----
pg_buffercache | 1.0     | public  | examine the shared buffer cache
plpgsql       | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)

postgres=# select name,setting,unit,current_setting(name) from pg_settings where
name='shared_buffers';
name          | setting | unit | current_setting
-----+-----+-----+-----
shared_buffers | 131072  | 8kB  | 1GB
(1 row)
```

14 May 2013

Following one of the suggestions from the PERFORM list, it seems that setting random_page_cost to 1 (from the default of 4) helps improve an actual query performed by the STOQS UI. With default random_page_cost = 4:

```
stoqs_march2013_s=# set random_page_cost=4;
SET
stoqs_march2013_s=# explain analyze SELECT stoqs_measuredparameter.id,
stoqs_parameter.name AS parameter__name,
stoqs_parameter.standard_name AS parameter__standard_name,
stoqs_measurement.depth AS measurement__depth,
stoqs_measurement.geom AS measurement__geom,
stoqs_instantpoint.timevalue AS measurement__instantpoint__timevalue,
stoqs_platform.name AS measurement__instantpoint__activity__platform__name,
stoqs_measuredparameter.datavalue AS datavalue,
stoqs_parameter.units AS parameter__units
FROM stoqs_parameter p1,
stoqs_measuredparameter
INNER JOIN stoqs_measurement ON (stoqs_measuredparameter.measurement_id = stoqs_measurement.id)
```

```

INNER JOIN stoqs_instantpoint ON (stoqs_measurement.instantpoint_id = stoqs_instantpoint.id)
INNER JOIN stoqs_parameter ON (stoqs_measuredparameter.parameter_id = stoqs_parameter.id)
INNER JOIN stoqs_activity ON (stoqs_instantpoint.activity_id = stoqs_activity.id)
INNER JOIN stoqs_platform ON (stoqs_activity.platform_id = stoqs_platform.id)
INNER JOIN stoqs_measuredparameter mp1 ON mp1.measurement_id =
stoqs_measuredparameter.measurement_id
WHERE (p1.name = 'sea_water_sigma_t')
AND (mp1.datavalue > 25.19)
AND (mp1.datavalue < 26.01)
AND (mp1.parameter_id = p1.id)
AND (stoqs_instantpoint.timevalue <= '2013-03-17 19:05:06'
AND stoqs_instantpoint.timevalue >= '2013-03-17 15:35:13'
AND stoqs_parameter.name IN ('fl700_uncorr')
AND stoqs_measurement.depth >= -1.88
AND stoqs_platform.name IN ('dorado')
AND stoqs_measurement.depth <= 83.57)
ORDER BY stoqs_activity.name ASC, stoqs_instantpoint.timevalue ASC;
QUERY PLAN

```

```

-----
Sort (cost=10741.41..10741.42 rows=1 width=1282) (actual time=358.452..358.452 rows=0 loops=1)
Sort Key: stoqs_activity.name, stoqs_instantpoint.timevalue
Sort Method: quicksort Memory: 25kB
-> Hash Join (cost=3002.89..10741.40 rows=1 width=1282) (actual time=358.445..358.445 rows=0
loops=1)
Hash Cond: (stoqs_instantpoint.activity_id = stoqs_activity.id)
-> Nested Loop (cost=2983.69..10722.19 rows=3 width=954) (actual time=358.269..358.269 rows=0
loops=1)
-> Nested Loop (cost=2983.69..9617.36 rows=191 width=946) (actual time=12.788..296.022 rows=20170
loops=1)
-> Hash Join (cost=2983.69..8499.07 rows=193 width=842) (actual time=12.778..223.137 rows=20170
loops=1)
Hash Cond: (stoqs_measuredparameter.parameter_id = stoqs_parameter.id)
-> Nested Loop (cost=2982.38..8478.47 rows=4628 width=24) (actual time=12.752..188.772
rows=197746 loops=1)
-> Nested Loop (cost=2982.38..4862.37 rows=512 width=4) (actual time=12.733..25.939 rows=20170
loops=1)
-> Seq Scan on stoqs_parameter p1 (cost=0.00..1.30 rows=1 width=4) (actual time=0.001..0.007
rows=1 loops=1)
Filter: ((name)::text = 'sea_water_sigma_t'::text)
-> Bitmap Heap Scan on stoqs_measuredparameter mp1 (cost=2982.38..4854.40 rows=534 width=8)
(actual time=12.724..21.223 rows=20170 loops
=1)
Recheck Cond: ((datavalue > 25.19::double precision) AND (datavalue < 26.01::double precision) AND
(parameter_id = p1.id))
-> BitmapAnd (cost=2982.38..2982.38 rows=534 width=0) (actual time=12.326..12.326 rows=0 loops=1)
-> Bitmap Index Scan on stoqs_measuredparameter_datavalue (cost=0.00..259.54 rows=12292 width=0)
(actual time=5.938..5.938 r
ows=23641 loops=1)
Index Cond: ((datavalue > 25.19::double precision) AND (datavalue < 26.01::double precision))
-> Bitmap Index Scan on stoqs_measuredparameter_parameter_id (cost=0.00..2719.38 rows=147035
width=0) (actual time=5.847..5.
847 rows=34750 loops=1)
Index Cond: (parameter_id = p1.id)
-> Index Scan using stoqs_measuredparameter_measurement_id on stoqs_measuredparameter
(cost=0.00..6.98 rows=7 width=20) (actual time=0.003..0.
005 rows=10 loops=20170)
Index Cond: (measurement_id = mp1.measurement_id)
-> Hash (cost=1.30..1.30 rows=1 width=826) (actual time=0.010..0.010 rows=1 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 1kB
-> Seq Scan on stoqs_parameter (cost=0.00..1.30 rows=1 width=826) (actual time=0.004..0.008
rows=1 loops=1)
Filter: ((name)::text = 'fl700_uncorr'::text)
-> Index Scan using stoqs_measurement_pkey on stoqs_measurement (cost=0.00..5.78 rows=1
width=116) (actual time=0.003..0.003 rows=1 loops=20170)

```

```

Index Cond: (id = stoqs_measuredparameter.measurement_id)
Filter: ((depth >= (-1.88)::double precision) AND (depth <= 83.57::double precision))
-> Index Scan using stoqs_instantpoint_pkey on stoqs_instantpoint (cost=0.00..5.77 rows=1
width=16) (actual time=0.003..0.003 rows=0 loops=20170)
Index Cond: (id = stoqs_measurement.instantpoint_id)
Filter: ((timevalue <= '2013-03-17 19:05:06-07'::timestamp with time zone) AND (timevalue >=
'2013-03-17 15:35:13-07'::timestamp with time zone))
-> Hash (cost=18.82..18.82 rows=30 width=336) (actual time=0.165..0.165 rows=7 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 1kB
-> Hash Join (cost=1.09..18.82 rows=30 width=336) (actual time=0.073..0.162 rows=7 loops=1)
Hash Cond: (stoqs_activity.platform_id = stoqs_platform.id)
-> Seq Scan on stoqs_activity (cost=0.00..16.77 rows=177 width=66) (actual time=0.006..0.048
rows=177 loops=1)
-> Hash (cost=1.07..1.07 rows=1 width=278) (actual time=0.018..0.018 rows=1 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 1kB
-> Seq Scan on stoqs_platform (cost=0.00..1.07 rows=1 width=278) (actual time=0.012..0.015 rows=1
loops=1)
Filter: ((name)::text = 'dorado'::text)
Total runtime: 358.698 ms
(42 rows)

```

With random_page_cost = 1:

```

stoqs_march2013_s=# set random_page_cost=1;
SET
stoqs_march2013_s=# explain analyze SELECT stoqs_measuredparameter.id,
stoqs_parameter.name AS parameter__name,
stoqs_parameter.standard_name AS parameter__standard_name,
stoqs_measurement.depth AS measurement__depth,
stoqs_measurement.geom AS measurement__geom,
stoqs_instantpoint.timevalue AS measurement__instantpoint__timevalue,
stoqs_platform.name AS measurement__instantpoint__activity__platform__name,
stoqs_measuredparameter.datavalue AS datavalue,
stoqs_parameter.units AS parameter__units
FROM stoqs_parameter p1,
stoqs_measuredparameter
INNER JOIN stoqs_measurement ON (stoqs_measuredparameter.measurement_id = stoqs_measurement.id)
INNER JOIN stoqs_instantpoint ON (stoqs_measurement.instantpoint_id = stoqs_instantpoint.id)
INNER JOIN stoqs_parameter ON (stoqs_measuredparameter.parameter_id = stoqs_parameter.id)
INNER JOIN stoqs_activity ON (stoqs_instantpoint.activity_id = stoqs_activity.id)
INNER JOIN stoqs_platform ON (stoqs_activity.platform_id = stoqs_platform.id)
INNER JOIN stoqs_measuredparameter mp1 ON mp1.measurement_id =
stoqs_measuredparameter.measurement_id
WHERE (p1.name = 'sea_water_sigma_t')
AND (mp1.datavalue > 25.19)
AND (mp1.datavalue < 26.01)
AND (mp1.parameter_id = p1.id)
AND (stoqs_instantpoint.timevalue <= '2013-03-17 19:05:06'
AND stoqs_instantpoint.timevalue >= '2013-03-17 15:35:13'
AND stoqs_parameter.name IN ('fl700_uncorr'))
AND stoqs_measurement.depth >= -1.88
AND stoqs_platform.name IN ('dorado')
AND stoqs_measurement.depth <= 83.57)
ORDER BY stoqs_activity.name ASC, stoqs_instantpoint.timevalue ASC;
QUERY PLAN

```

```

-----
Sort (cost=3930.09..3930.09 rows=1 width=1282) (actual time=185.676..185.676 rows=0 loops=1)
Sort Key: stoqs_activity.name, stoqs_instantpoint.timevalue
Sort Method: quicksort Memory: 25kB
-> Hash Join (cost=1678.86..3930.08 rows=1 width=1282) (actual time=185.656..185.656 rows=0
loops=1)
Hash Cond: (stoqs_measuredparameter.parameter_id = stoqs_parameter.id)
-> Nested Loop (cost=1677.55..3928.72 rows=10 width=464) (actual time=185.626..185.626 rows=0
loops=1)

```

```

-> Hash Join (cost=1677.55..3926.60 rows=1 width=456) (actual time=185.625..185.625 rows=0
loops=1)
Hash Cond: (stoqs_instantpoint.activity_id = stoqs_activity.id)
-> Nested Loop (cost=1667.82..3916.85 rows=7 width=128) (actual time=185.548..185.548 rows=0
loops=1)
-> Nested Loop (cost=1667.82..3075.30 rows=507 width=120) (actual time=12.112..111.871 rows=20170
loops=1)
-> Nested Loop (cost=1667.82..2211.81 rows=512 width=4) (actual time=12.098..26.814 rows=20170
loops=1)
-> Seq Scan on stoqs_parameter p1 (cost=0.00..1.30 rows=1 width=4) (actual time=0.003..0.011
rows=1 loops=1)
Filter: ((name)::text = 'sea_water_sigma_t'::text)
-> Bitmap Heap Scan on stoqs_measuredparameter mp1 (cost=1667.82..2203.83 rows=534 width=8)
(actual time=12.088..20.804 rows=20170 loops
=1)
Recheck Cond: ((datavalue > 25.19::double precision) AND (datavalue < 26.01::double precision) AND
(parameter_id = p1.id))
-> BitmapAnd (cost=1667.82..1667.82 rows=534 width=0) (actual time=11.707..11.707 rows=0 loops=1)
-> Bitmap Index Scan on stoqs_measuredparameter_datavalue (cost=0.00..157.26 rows=12292 width=0)
(actual time=5.885..5.885 rows=23641 loops=1)
Index Cond: ((datavalue > 25.19::double precision) AND (datavalue < 26.01::double precision))
-> Bitmap Index Scan on stoqs_measuredparameter_parameter_id (cost=0.00..1507.11 rows=147035
width=0) (actual time=5.265..5.265 rows=34750 loops=1)
Index Cond: (parameter_id = p1.id)
-> Index Scan using stoqs_measurement_pkey on stoqs_measurement (cost=0.00..1.67 rows=1
width=116) (actual time=0.003..0.003 rows=1 loops=2017
0)
Index Cond: (id = mp1.measurement_id)
Filter: ((depth >= (-1.88)::double precision) AND (depth <= 83.57::double precision))
-> Index Scan using stoqs_instantpoint_pkey on stoqs_instantpoint (cost=0.00..1.65 rows=1
width=16) (actual time=0.003..0.003 rows=0 loops=20170)
Index Cond: (id = stoqs_measurement.instantpoint_id)
Filter: ((timevalue <= '2013-03-17 19:05:06-07'::timestamp with time zone) AND (timevalue >=
'2013-03-17 15:35:13-07'::timestamp with time zone)
)
-> Hash (cost=9.35..9.35 rows=30 width=336) (actual time=0.071..0.071 rows=7 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 1kB
-> Nested Loop (cost=0.00..9.35 rows=30 width=336) (actual time=0.057..0.064 rows=7 loops=1)
-> Seq Scan on stoqs_platform (cost=0.00..1.07 rows=1 width=278) (actual time=0.004..0.006 rows=1
loops=1)
Filter: ((name)::text = 'dorado'::text)
-> Index Scan using stoqs_activity_platform_id on stoqs_activity (cost=0.00..7.83 rows=35
width=66) (actual time=0.049..0.052 rows=7 loops=1)
Index Cond: (platform_id = stoqs_platform.id)
-> Index Scan using stoqs_measuredparameter_measurement_id on stoqs_measuredparameter
(cost=0.00..2.02 rows=7 width=20) (never executed)
Index Cond: (measurement_id = stoqs_measurement.id)
-> Hash (cost=1.30..1.30 rows=1 width=826) (actual time=0.023..0.023 rows=1 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 1kB
-> Seq Scan on stoqs_parameter (cost=0.00..1.30 rows=1 width=826) (actual time=0.016..0.020
rows=1 loops=1)
Filter: ((name)::text = 'fl700_uncorr'::text)
Total runtime: 185.850 ms
(40 rows)

```

Query went from 0.36 seconds to 0.18 seconds.

Based on this the random_page_cost setting in postgresql.conf was changed from 4 to 1.01 on pgtest. Comparing similar queries between http://odss-test.shore.mbari.org/canon/stoqs_march2013_s_pgtest/query/ and http://odss-test.shore.mbari.org/canon/stoqs_march2013_s/query/ shows no appreciable improvement in performance on the pgtest database.

15 May 2013

Preparing for when the STOQS application will be setting work_mem for the queries that need it I have set up logging of temp files that are created when there is not enough work_mem for sorts to be done in memory. The instructions are in the book and on this page:

http://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server

PostgreSQL 8.3 and later

In 8.3 you can use log_temp_files to figure out if sorts are using disk instead of fitting in memory. In earlier versions, you might instead just monitor the size of them by looking at how much space is being used in the various \$PGDATA/base/<db oid>/pgsql_tmp files.

You can see sorts to disk happen in EXPLAIN ANALYZE plans as well. For example, if you see a line like "Sort Method: external merge

Disk: 7526kB" in there, you'd know a work_mem of at least 8MB would really improve how fast that query executed, by sorting in RAM instead of swapping to disk.

The change to the postgresql.conf file was made today with the previous version saved to postgresql.conf.130515

16 May 2013

Another post to the pgsq-l-perform mail list asking for opinions on a more conventional server configuration:

From: Mike McCann <mccann@mbari.org>
Subject: Re: [PERFORM] Hardware suggestions for maximum read performance
Date: May 13, 2013 3:36:03 PM PDT
To: pgsq-l-performance@postgresql.org
Cc: Yuri Levinsky <yuril@celtick.com>, Scott Marlowe <scott.marlowe@gmail.com>, Julien Cigar <jcigar@ulb.ac.be>, Arjen van der Meijden <acmmailing@tweakers.net>, Jeff Janes <jeff.janes@gmail.com>

On May 7, 2013, at 4:21 PM, Jeff Janes wrote:

On Thu, May 2, 2013 at 6:35 PM, Scott Marlowe <scott.marlowe@gmail.com> wrote:

On Thu, May 2, 2013 at 5:11 PM, Mike McCann <mccann@mbari.org> wrote:

> Hello,

>

> We are in the fortunate situation of having more money than time to help
> solve our PostgreSQL 9.1 performance problem.

>

> Our server hosts databases that are about 1 GB in size with the largest
> tables having order 10 million 20-byte indexed records. The data are loaded
> once and then read from a web app and other client programs. Some of the
> queries execute ORDER BY on the results. There are typically less than a
> dozen read-only concurrent connections to any one database.

I wouldn't count on this being a problem that can be fixed merely by throwing money at it.

How many rows does any one of these queries need to access and then ORDER BY?

...

>

> HP ProLiant DL360p Gen 8
> Dual Intel Xeon 2.4GHz 4-core E5-2609 CPUs
> 64GB RAM
> 2x146GB 15K SAS hard drives
> 3x200GB SATA SLC SSDs
> + the usual accessories (optical drive, rail kit, dual power supplies)

If your DB is 1G, and will grow to 10G then the IO shouldn't be any

problem, as the whole db should be cached in memory.

But it can take a surprisingly long time to get it cached in the first place, from a cold start.

If that is the problem, pg_prewarm could help.

Cheers,

Jeff

Thank you everyone for your suggestions.

It's clear that our current read performance was not limited by hardware. An 'explain analyze' for a sample query is:

```
stoqs_march2013_s=# show work_mem;
work_mem
-----
1MB
(1 row)
```

```
stoqs_march2013_s=# explain analyze select * from stoqs_measuredparameter order by datavalue;
QUERY PLAN
-----
Sort (cost=541002.15..549456.68 rows=3381814 width=20) (actual time=6254.780..7244.074
rows=3381814 loops=1)
Sort Key: datavalue
Sort Method: external merge  Disk: 112424kB
-> Seq Scan on stoqs_measuredparameter (cost=0.00..55359.14 rows=3381814 width=20) (actual
time=0.011..354.385 rows=3381814 loops=1)
Total runtime: 7425.854 ms
(5 rows)
```

Increasing work_mem to 355 MB improves the performance by a factor of 2:

```
stoqs_march2013_s=# set work_mem='355MB';
SET
stoqs_march2013_s=# explain analyze select * from stoqs_measuredparameter order by datavalue;
QUERY PLAN
-----
Sort (cost=422106.15..430560.68 rows=3381814 width=20) (actual time=2503.078..2937.130
rows=3381814 loops=1)
Sort Key: datavalue
Sort Method: quicksort  Memory: 362509kB
-> Seq Scan on stoqs_measuredparameter (cost=0.00..55359.14 rows=3381814 width=20) (actual
time=0.016..335.745 rows=3381814 loops=1)
Total runtime: 3094.601 ms
(5 rows)
```

I tried changing random_page_cost to from 4 to 1 and saw no change.

I'm wondering now what changes might get this query to run in less than one second. If all the data is in memory, then will faster CPU and memory be the things that help?

We have an alternate (a bit more conventional) server configuration that we are considering:

HP ProLiant DL360p Gen 8
Dual Intel Xeon 3.3GHz 4-core E5-2643 CPUs
128GB PC3-12800 RAM
16x146GB 15K SAS hard drives
HP Smart Array P822/2GB FBWC controller + P420i w/ 2GB FBWC
+ the usual accessories (optical drive, rail kit, dual power supplies)

All suggestions welcomed!

-Mike

--

Mike McCann
Software Engineer
Monterey Bay Aquarium Research Institute
7700 Sandholdt Road
Moss Landing, CA 95039-9644
Voice: 831.775.1769 Fax: 831.775.1736 <http://www.mbari.org>

The responses:

From: Mike McCann <mccann@mbari.org>
Subject: Re: [PERFORM] Hardware suggestions for maximum read performance
Date: May 13, 2013 4:58:17 PM PDT
To: Jeff Janes <jeff.janes@gmail.com>
Cc: "pgsql-performance@postgresql.org" <pgsql-performance@postgresql.org>, Yuri Levinsky <yuril@celltick.com>, Scott Marlowe <scott.marlowe@gmail.com>, Julien Cigar <jcigar@ulb.ac.be>, Arjen van der Meijden <acmmailing@tweakers.net>

On May 13, 2013, at 4:24 PM, Jeff Janes wrote:

```
> On Mon, May 13, 2013 at 3:36 PM, Mike McCann <mccann@mbari.org> wrote:
>
> Increasing work_mem to 355 MB improves the performance by a factor of 2:
>
> stoqs_march2013_s=# set work_mem='355MB';
> SET
> stoqs_march2013_s=# explain analyze select * from stoqs_measuredparameter order by datavalue;
>                                     QUERY PLAN
>
> -----
> Sort (cost=422106.15..430560.68 rows=3381814 width=20) (actual time=2503.078..2937.130
rows=3381814 loops=1)
>   Sort Key: datavalue
>   Sort Method: quicksort  Memory: 362509kB
>   -> Seq Scan on stoqs_measuredparameter (cost=0.00..55359.14 rows=3381814 width=20) (actual
time=0.016..335.745 rows=3381814 loops=1)
> Total runtime: 3094.601 ms
> (5 rows)
>
> I tried changing random_page_cost to from 4 to 1 and saw no change.
>
> I'm wondering now what changes might get this query to run in less than one second.
```

I think you are worrying about the wrong thing here. What is a web app going to do with 3,381,814 rows, once it obtains them? Your current testing is not testing the time it takes to stream that data to the client, or for the client to do something meaningful with that data.

If you only plan to actually fetch a few dozen of those rows, then you probably need to incorporate that into your test, either by using a LIMIT, or by using a mock-up of the actual application to do some timings.

Also, what is the type and collation of the column you are sorting on? non-'C' collations of text columns sort about 3 times slower than 'C' collation does.

If all the data is in memory, then will faster CPU and memory be the things that help?

Yes, those would help (it is not clear to me which of the two would help more), but I think you need to rethink your design of sending the entire database table to the application server for each page-view.

Cheers,

Jeff

Hi Jeff,

The datavalue column is double precision:

```
stoqs_march2013_s=# \d+ stoqs_measuredparameter
Table "public.stoqs_measuredparameter"
Column          |          Type          |          Modifiers
| Storage | Description
-----+-----
id               | integer                | not null default
nextval('stoqs_measuredparameter_id_seq'::regclass) | plain |
measurement_id  | integer                | not null
| plain |
parameter_id    | integer                | not null
| plain |
datavalue       | double precision       | not null
| plain |
Indexes:
"stoqs_measuredparameter_pkey" PRIMARY KEY, btree (id)
"stoqs_measuredparameter_measurement_id_parameter_id_key" UNIQUE CONSTRAINT, btree (measurement_id,
parameter_id)
"stoqs_measuredparameter_datavalue" btree (datavalue)
"stoqs_measuredparameter_measurement_id" btree (measurement_id)
"stoqs_measuredparameter_parameter_id" btree (parameter_id)
Foreign-key constraints:
"stoqs_measuredparameter_measurement_id_fkey" FOREIGN KEY (measurement_id) REFERENCES
stoqs_measurement(id) DEFERRABLE INITIALLY DEFERRED
"stoqs_measuredparameter_parameter_id_fkey" FOREIGN KEY (parameter_id) REFERENCES
stoqs_parameter(id) DEFERRABLE INITIALLY DEFERRED
Has OIDs: no
```

Thanks for the suggestion and advice to examine the web app performance. We've actually taken quite a few steps to optimize how the web app works. The example query I provided is a simple worst-case one that we can use to help us decide on the proper hardware. An actual query performed by the web app is:

```
stoqs_march2013_s=# explain analyze SELECT stoqs_measuredparameter.id,
stoqs_march2013_s=#         stoqs_parameter.name AS parameter__name,
stoqs_march2013_s=#         stoqs_parameter.standard_name AS parameter__standard_name,
stoqs_march2013_s=#         stoqs_measurement.depth AS measurement__depth,
stoqs_march2013_s=#         stoqs_measurement.geom AS measurement__geom,
stoqs_march2013_s=#         stoqs_instantpoint.timevalue AS measurement__instantpoint__timevalue,
stoqs_march2013_s=#         stoqs_platform.name AS
measurement__instantpoint__activity__platform__name,
stoqs_march2013_s=#         stoqs_measuredparameter.datavalue AS datavalue,
stoqs_march2013_s=#         stoqs_parameter.units AS parameter__units
stoqs_march2013_s=# FROM stoqs_parameter p1,
stoqs_march2013_s=#         stoqs_measuredparameter
stoqs_march2013_s=# INNER JOIN stoqs_measurement ON (stoqs_measuredparameter.measurement_id =
stoqs_measurement.id)
stoqs_march2013_s=# INNER JOIN stoqs_instantpoint ON (stoqs_measurement.instantpoint_id =
stoqs_instantpoint.id)
stoqs_march2013_s=# INNER JOIN stoqs_parameter ON (stoqs_measuredparameter.parameter_id =
stoqs_parameter.id)
```

```

stoqs_march2013_s-#      INNER JOIN stoqs_activity ON (stoqs_instantpoint.activity_id =
stoqs_activity.id)
stoqs_march2013_s-#      INNER JOIN stoqs_platform ON (stoqs_activity.platform_id =
stoqs_platform.id)
stoqs_march2013_s-#      INNER JOIN stoqs_measuredparameter mp1 ON mp1.measurement_id =
stoqs_measuredparameter.measurement_id
stoqs_march2013_s-# WHERE (p1.name = 'sea_water_sigma_t')
stoqs_march2013_s-#      AND (mp1.datavalue > 25.19)
stoqs_march2013_s-#      AND (mp1.datavalue < 26.01)
stoqs_march2013_s-#      AND (mp1.parameter_id = p1.id)
stoqs_march2013_s-#      AND (stoqs_instantpoint.timevalue <= '2013-03-17 19:05:06'
stoqs_march2013_s-#      AND stoqs_instantpoint.timevalue >= '2013-03-17 15:35:13'
stoqs_march2013_s-#      AND stoqs_parameter.name IN ('fl700_uncorr')
stoqs_march2013_s-#      AND stoqs_measurement.depth >= -1.88
stoqs_march2013_s-#      AND stoqs_platform.name IN ('dorado')
stoqs_march2013_s-#      AND stoqs_measurement.depth <= 83.57)
stoqs_march2013_s-# ORDER BY stoqs_activity.name ASC, stoqs_instantpoint.timevalue ASC;
QUERY PLAN

```

```

-----
Sort (cost=10741.41..10741.42 rows=1 width=1282) (actual time=770.211..770.211 rows=0 loops=1)
Sort Key: stoqs_activity.name, stoqs_instantpoint.timevalue
Sort Method: quicksort Memory: 25kB
-> Hash Join (cost=3002.89..10741.40 rows=1 width=1282) (actual time=770.200..770.200 rows=0
loops=1)
Hash Cond: (stoqs_instantpoint.activity_id = stoqs_activity.id)
-> Nested Loop (cost=2983.69..10722.19 rows=3 width=954) (actual time=770.036..770.036 rows=0
loops=1)
-> Nested Loop (cost=2983.69..9617.36 rows=191 width=946) (actual time=91.369..680.072 rows=20170
loops=1)
-> Hash Join (cost=2983.69..8499.07 rows=193 width=842) (actual time=91.346..577.633 rows=20170
loops=1)
Hash Cond: (stoqs_measuredparameter.parameter_id = stoqs_parameter.id)
-> Nested Loop (cost=2982.38..8478.47 rows=4628 width=24) (actual time=91.280..531.408
rows=197746 loops=1)
-> Nested Loop (cost=2982.38..4862.37 rows=512 width=4) (actual time=91.202..116.140 rows=20170
loops=1)
-> Seq Scan on stoqs_parameter p1 (cost=0.00..1.30 rows=1 width=4) (actual time=0.002..0.011
rows=1 loops=1)
Filter: ((name)::text = 'sea_water_sigma_t'::text)
-> Bitmap Heap Scan on stoqs_measuredparameter mp1 (cost=2982.38..4854.40 rows=534 width=8)
(actual time=91.194..109.846 rows=20170 loop
s=1)
Recheck Cond: ((datavalue > 25.19::double precision) AND (datavalue < 26.01::double precision) AND
(parameter_id = p1.id))
-> BitmapAnd (cost=2982.38..2982.38 rows=534 width=0) (actual time=90.794..90.794 rows=0 loops=1)
-> Bitmap Index Scan on stoqs_measuredparameter_datavalue (cost=0.00..259.54 rows=12292 width=0)
(actual time=62.769..62.769
rows=23641 loops=1)
Index Cond: ((datavalue > 25.19::double precision) AND (datavalue < 26.01::double precision))
-> Bitmap Index Scan on stoqs_measuredparameter_parameter_id (cost=0.00..2719.38 rows=147035
width=0) (actual time=27.412..2
7.412 rows=34750 loops=1)
Index Cond: (parameter_id = p1.id)
-> Index Scan using stoqs_measuredparameter_measurement_id on stoqs_measuredparameter
(cost=0.00..6.98 rows=7 width=20) (actual time=0.008..0.
017 rows=10 loops=20170)
Index Cond: (measurement_id = mp1.measurement_id)
-> Hash (cost=1.30..1.30 rows=1 width=826) (actual time=0.012..0.012 rows=1 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 1kB
-> Seq Scan on stoqs_parameter (cost=0.00..1.30 rows=1 width=826) (actual time=0.007..0.010
rows=1 loops=1)
Filter: ((name)::text = 'fl700_uncorr'::text)
-> Index Scan using stoqs_measurement_pkey on stoqs_measurement (cost=0.00..5.78 rows=1
width=116) (actual time=0.004..0.004 rows=1 loops=20170)

```

```

Index Cond: (id = stoqs_measuredparameter.measurement_id)
Filter: ((depth >= (-1.88)::double precision) AND (depth <= 83.57::double precision))
-> Index Scan using stoqs_instantpoint_pkey on stoqs_instantpoint (cost=0.00..5.77 rows=1
width=16) (actual time=0.004..0.004 rows=0 loops=20170)
Index Cond: (id = stoqs_measurement.instantpoint_id)
Filter: ((timevalue <= '2013-03-17 19:05:06-07'::timestamp with time zone) AND (timevalue >=
'2013-03-17 15:35:13-07'::timestamp with time zone))
-> Hash (cost=18.82..18.82 rows=30 width=336) (actual time=0.151..0.151 rows=7 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 1kB
-> Hash Join (cost=1.09..18.82 rows=30 width=336) (actual time=0.035..0.145 rows=7 loops=1)
Hash Cond: (stoqs_activity.platform_id = stoqs_platform.id)
-> Seq Scan on stoqs_activity (cost=0.00..16.77 rows=177 width=66) (actual time=0.005..0.069
rows=177 loops=1)
-> Hash (cost=1.07..1.07 rows=1 width=278) (actual time=0.014..0.014 rows=1 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 1kB
-> Seq Scan on stoqs_platform (cost=0.00..1.07 rows=1 width=278) (actual time=0.008..0.012 rows=1
loops=1)
Filter: ((name)::text = 'dorado'::text)
Total runtime: 770.445 ms
(42 rows)

```

We assume that steps taken to improve the worst-case query scenario will also improve these kind of queries. If anything above pops out as needing better planning please let us know that too!

Thanks,
Mike

```

--
Mike McCann
Software Engineer
Monterey Bay Aquarium Research Institute
7700 Sandholdt Road
Moss Landing, CA 95039-9644
Voice: 831.775.1769 Fax: 831.775.1736 http://www.mbari.org

```

Another response:

```

From: Scott Marlowe <scott.marlowe@gmail.com>
Subject: Re: [PERFORM] Hardware suggestions for maximum read performance
Date: May 13, 2013 6:09:58 PM PDT
To: Mike McCann <mccann@mbari.org>
Cc: Jeff Janes <jeff.janes@gmail.com>, "pgsql-performance@postgresql.org" <pgsql-
performance@postgresql.org>, Yuri Levinsky <yuril@celltick.com>,
Julien Cigar <jcigar@ulb.ac.be>, Arjen van der Meijden <acmmailing@tweakers.net>

```

On Mon, May 13, 2013 at 5:58 PM, Mike McCann <mccann@mbari.org> wrote:

```

> We assume that steps taken to improve the worst-case query scenario will
> also improve these kind of queries. If anything above pops out as needing
> better planning please let us know that too!

```

Bad assumption. If your real workload will be queries like the one here that takes 700 ms, but you'll be running 10,000 of them a second, you're tuning / hardware choices are going to be much different then if your query is going to be the previous 7 second one. Use realistic queries, not ones that are nothing like what your real ones will be. then use pgbench and its ability to run custom sql scripts to get a REAL idea how your hardware performs. Note that if you will run the slow query you posted like once a minute and roll it up or cache it then don't get too worried about it. Pay attention to the queries that will add up, in aggregate, to your greatest load.

Not a whole lot of opinions except that tuning a server requires diligent work and performance is something one must do some work to achieve, and cannot be had by simply throwing money at the problem.

20 May 2013

A few more responses over the weekend:

From: Greg Smith <greg@2ndQuadrant.com>
Subject: Re: [PERFORM] Hardware suggestions for maximum read performance
Date: May 19, 2013 7:44:16 PM PDT
To: Mike McCann <mccann@mbari.org>
Cc: pgsql-performance@postgresql.org, Yuri Levinsky <yuril@celltick.com>, Scott Marlowe <scott.marlowe@gmail.com>, Julien Cigar <jcigar@ulb.ac.be>, Arjen van der Meijden <acmmailing@tweakers.net>, Jeff Janes <jeff.janes@gmail.com>

On 5/13/13 6:36 PM, Mike McCann wrote:

```
> stoqs_march2013_s=# explain analyze select * from
> stoqs_measuredparameter order by datavalue;
>
> QUERY PLAN
>
```

```
-----
> Sort (cost=422106.15..430560.68 rows=3381814 width=20) (actual
> time=2503.078..2937.130 rows=3381814 loops=1)
>   Sort Key: datavalue
>   Sort Method: quicksort  Memory: 362509kB
>   -> Seq Scan on stoqs_measuredparameter (cost=0.00..55359.14
> rows=3381814 width=20) (actual time=0.016..335.745 rows=3381814 loops=1)
>     Total runtime: 3094.601 ms
> (5 rows)
```

```
> I tried changing random_page_cost to from 4 to 1 and saw no change.
```

Have you tried putting an index by datavalue on this table? Once you've done that, then changing random_page_cost will make using that index look less expensive. Sorting chews through a good bit of CPU time, and that's where all of your runtime is being spent at--once you increase work_mem up very high that is.

```
> I'm wondering now what changes might get this query to run in less than
> one second. If all the data is in memory, then will faster CPU and
> memory be the things that help?
```

You're trying to fix a fundamental design issue with hardware. That usually doesn't go well. Once you get a box big enough to hold the whole database in RAM, beyond that the differences between server systems are relatively small.

--

Greg Smith 2ndQuadrant US greg@2ndQuadrant.com Baltimore, MD
PostgreSQL Training, Services, and 24x7 Support www.2ndQuadrant.com

and

From: Scott Marlowe <scott.marlowe@gmail.com>
Subject: Re: [PERFORM] Hardware suggestions for maximum read performance
Date: May 19, 2013 8:57:32 PM PDT
To: Greg Smith <greg@2ndquadrant.com>
Cc: Mike McCann <mccann@mbari.org>, "pgsql-performance@postgresql.org" <pgsql-performance@postgresql.org>, Yuri Levinsky <yuril@celltick.com>, Julien Cigar <jcigar@ulb.ac.be>, Arjen van der Meijden <acmmailing@tweakers.net>, Jeff Janes <jeff.janes@gmail.com>

On Sun, May 19, 2013 at 8:44 PM, Greg Smith <greg@2ndquadrant.com> wrote:

```
>On 5/13/13 6:36 PM, Mike McCann wrote:
>>
>> stoqs_march2013_s=# explain analyze select * from
>> stoqs_measuredparameter order by datavalue;
>>
>> QUERY PLAN
>>
```

```
>>-----
>> Sort (cost=422106.15..430560.68 rows=3381814 width=20) (actual
>> time=2503.078..2937.130 rows=3381814 loops=1)
>>   Sort Key: datavalue
>>   Sort Method: quicksort  Memory: 362509kB
>>   -> Seq Scan on stoqs_measuredparameter (cost=0.00..55359.14
>> rows=3381814 width=20) (actual time=0.016..335.745 rows=3381814 loops=1)
>>   Total runtime: 3094.601 ms
>> (5 rows)
>>
>> I tried changing random_page_cost to from 4 to 1 and saw no change.
>
>
> Have you tried putting an index by datavalue on this table? Once you've
> done that, then changing random_page_cost will make using that index look
> less expensive. Sorting chews through a good bit of CPU time, and that's
> where all of your runtime is being spent at--once you increase work_mem up
> very high that is.
```

This++ plus cluster on that index if you can.

2 July 2013

We made the decision made to purchase a server with a FusionIO card. Server is now set up on the network with name "kraken". Here is the [confluence page for kraken's configuration settings](#).

Setting up ability to run pgbench-tools on all the various servers. This may be done per user account by first installing from git and following the instructions in README.rst; however, the scripts need to run commands requiring createdb and vacuum privileges, so it's best to run as user postgres. So, logging into pgtest as user postgres and getting things set up:

```
-bash-4.1$ pwd
/var/lib/pgsql
-bash-4.1$ mkdir dev
-bash-4.1$ cd dev
-bash-4.1$ git clone git://git.postgresql.org/git/pgbench-tools.git
Initialized empty Git repository in /var/lib/pgsql/dev/pgbench-tools/.git/
remote: Counting objects: 108, done.
remote: Compressing objects: 100% (106/106), done.
remote: Total 108 (delta 54), reused 0 (delta 0)
Receiving objects: 100% (108/108), 24.61 KiB, done.
Resolving deltas: 100% (54/54), done.

-bash-4.1$ createdb results
-bash-4.1$ createdb pgbench

-bash-4.1$ cd pgbench-tools/
-bash-4.1$ psql -f init/resultdb.sql -d results
psql:init/resultdb.sql:1: ERROR: table "timing" does not exist
CREATE TABLE
CREATE INDEX
psql:init/resultdb.sql:12: ERROR: table "tests" does not exist
psql:init/resultdb.sql:29: NOTICE: CREATE TABLE will create implicit sequence "tests_test_seq" for
serial column "tests.test"
psql:init/resultdb.sql:29: NOTICE: CREATE TABLE will create implicit sequence "tests_set_seq" for
serial column "tests.set"
CREATE TABLE
psql:init/resultdb.sql:31: ERROR: table "testset" does not exist
psql:init/resultdb.sql:36: NOTICE: CREATE TABLE will create implicit sequence "testset_set_seq"
for serial column "testset.set"
CREATE TABLE
INSERT 0 1
psql:init/resultdb.sql:40: ERROR: table "test_bgwriter" does not exist
CREATE TABLE

-bash-4.1$ psql -c "UPDATE testset SET info='Initial test set' WHERE set=1" -d results
```

UPDATE 1

I think those ERROR messages can be ignored because the following tests work.

The gnuplot package is needed for producing graphs of the results:

```
sudo -c "yum install gnuplot"
```

Edit the config file with hard path to pgbench and reduced number of scale and client runs so that we have a quick execution to just test things:

```
-bash-4.1$ mv config config.130702
-bash-4.1$ cp config.130702 config
-bash-4.1$ vi config
-bash-4.1$ diff config.130702 config
8c8
< PGBENCHBIN=`which pgbench`
---
> PGBENCHBIN=/usr/pgsql-9.1/bin/pgbench
42,44c42,47
< SCALES="1 10 100 1000"
< SETCLIENTS="1 2 4 8 16 32"
< SETTIMES=3
---
> SCALES="1"
> SETCLIENTS="1"
> SETTIMES=1
> #SCALES="1 10 100 1000"
> #SETCLIENTS="1 2 4 8 16 32"
> #SETTIMES=3
```

Edit scripts to use "local socket" connection (removing "-h \$TESTHOST") for the postgres user:

```
-bash-4.1$ diff runset.130702 runset
19c19
< TESTPSQL="psql -h $TESTHOST -U $TESTUSER -p $TESTPORT -d $TESTDB"
---
> TESTPSQL="psql -U $TESTUSER -p $TESTPORT -d $TESTDB"
24c24
< $PGBENCHBIN -i -s $SCALE -h $TESTHOST -U $TESTUSER -p $TESTPORT $TESTDB
---
> $PGBENCHBIN -i -s $SCALE -U $TESTUSER -p $TESTPORT $TESTDB

-bash-4.1$ diff benchwarmer.130702 benchwarmer
84,85c84,85
< TESTPSQL="psql -h $TESTHOST -U $TESTUSER -p $TESTPORT -d $TESTDB"
< RESULTPSQL="psql -h $RESULTHOST -U $RESULTUSER -p $RESULTPORT -d $RESULTDB"
---
> TESTPSQL="psql -U $TESTUSER -p $TESTPORT -d $TESTDB"
> RESULTPSQL="psql -U $RESULTUSER -p $RESULTPORT -d $RESULTDB"
158c158
< $PGBENCHBIN -f $BASEDIR/$TESTDIR/$SCRIPT -s $SCALE -l -n $TESTLEN -U $TESTUSER -h $TESTHOST -p
$TESTPORT -c $CLIENTS $WORKERS $TESTDB > results.txt &
---
> $PGBENCHBIN -f $BASEDIR/$TESTDIR/$SCRIPT -s $SCALE -l -n $TESTLEN -U $TESTUSER -p $TESTPORT -c
$CLIENTS $WORKERS $TESTDB > results.txt &

-bash-4.1$ diff webreport.130702 webreport
10c10
< RESULTPSQL="psql -h $RESULTHOST -U $RESULTUSER -p $RESULTPORT -d $RESULTDB"
---
> RESULTPSQL="psql -U $RESULTUSER -p $RESULTPORT -d $RESULTDB"
```

And now a test run looks like:

```
-bash-4.1$ ./runset
Removing old pgbench tables
```

```

DROP TABLE
VACUUM
Creating new pgbench tables
NOTICE: table "pgbench_branches" does not exist, skipping
NOTICE: table "pgbench_tellers" does not exist, skipping
NOTICE: table "pgbench_accounts" does not exist, skipping
NOTICE: table "pgbench_history" does not exist, skipping
creating tables...
10000 tuples done.
20000 tuples done.
30000 tuples done.
40000 tuples done.
50000 tuples done.
60000 tuples done.
70000 tuples done.
80000 tuples done.
90000 tuples done.
100000 tuples done.
set primary key...
NOTICE: ALTER TABLE / ADD PRIMARY KEY will create implicit index "pgbench_branches_pkey" for table
"pgbench_branches"
NOTICE: ALTER TABLE / ADD PRIMARY KEY will create implicit index "pgbench_tellers_pkey" for table
"pgbench_tellers"
NOTICE: ALTER TABLE / ADD PRIMARY KEY will create implicit index "pgbench_accounts_pkey" for table
"pgbench_accounts"
vacuum...done.
Run set #1 of 1 with 1 clients scale=1
Running tests using: psql -U postgres -p 5432 -d pgbench
Storing results using: psql -U postgres -p 5432 -d results
Cleaning up database pgbench
TRUNCATE TABLE
VACUUM
CHECKPOINT
INSERT 0 1
This is test 3
Script select.sql executing for 1 concurrent users...
UPDATE 1
transaction type: Custom query
scaling factor: 1
query mode: simple
number of clients: 1
number of threads: 1
duration: 60 s
number of transactions actually processed: 547599
tps = 9126.622772 (including connections establishing)
tps = 9127.071214 (excluding connections establishing)
Worst latency results:
2018
2311
2370
3155
3476
Could not find/open font when opening font "arial", using internal non-scalable font
Could not find/open font when opening font "arial", using internal non-scalable font

Could not find/open font when opening font "arial", using internal non-scalable font
Warning: empty x range [1:1], adjusting to [0.99:1.01]
Warning: empty y range [9228:9228], adjusting to [9135.72:9320.28]
Warning: empty y2 range [21:21], adjusting to [20.79:21.21]
Could not find/open font when opening font "arial", using internal non-scalable font
Warning: empty x range [1:1], adjusting to [0.99:1.01]
Warning: empty y range [9228:9228], adjusting to [9135.72:9320.28]
Could not find/open font when opening font "arial", using internal non-scalable font
Warning: empty x range [1:1], adjusting to [0.99:1.01]
Warning: empty y range [1:1], adjusting to [0.99:1.01]
Warning: empty z range [9228:9228], adjusting to [9135.72:9320.28]

```

3 July 2013

Making the results visible on an internally accessible web page. Turn on httpd and make an alias to the results directory that is produced above:

```
sudo /bin/bash
chkconfig httpd on
/sbin/service httpd start
```

Confirm that <http://pgtest.shore.mbari.org/> is visible. It wasn't because the firewall on pgtest was preventing it. A call to Pat Allen fixed this. Next, edit /etc/httpd/conf/httpd.conf to add the alias and make the postgres home dir world executable (don't do this on a production system!) so that the apache httpd process can serve the directory:

```
# diff httpd.conf.130703 httpd.conf
559a560,569
> # Expose our pgbench-toolbox results
> Alias /pgbench /var/lib/pgsql/dev/pgbench-tools/results
>
> <Directory "/var/lib/pgsql/dev/pgbench-tools/results">
>     Options Indexes FollowSymLinks
>     AllowOverride None
>     Order allow,deny
>     Allow from all
> </Directory>
>

# chmod +x /var/lib/pgsql
```

The pgbench-toolbox results can now be shared at <http://pgtest.shore.mbari.org/pgbench/>

Comments

Edit config file to change checkpoint_segments and checkpoint_completion_target

```
-bash-4.1$ diff postgresql.conf postgresql.conf.130703
178c178
< checkpoint_segments = 10          # in logfile segments, min 1, 16MB each
---
> #checkpoint_segments = 3          # in logfile segments, min 1, 16MB each
180c180
< checkpoint_completion_target = 0.9 # checkpoint target duration, 0.0 - 1.0
---
> #checkpoint_completion_target = 0.5      # checkpoint target duration, 0.0 - 1.0
```

Changed due to log

LOG: checkpoints are occurring too frequently (2 seconds apart)
HINT: Consider increasing the configuration parameter "checkpoint_segments".

Posted by [drolicheck](#) at Jul 03, 2013.

July 17 2013

Install pgfouine for log analyzer

```
sudo -c "yum install pgfouine"
```

Change date.timezone in /etc/php.ini file, errors will arise if this is left at default value.

```
-bash-4.1$ diff php.ini php.ini.130717
946c946
< date.timezone = America/Los_Angeles
```

```
---
> ;date.timezone =
```

Run a pgbench test to generate some query traffic.

```
-bash-4.1$ ./pgbench -T 60 -c 4 pgbench
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 160
query mode: simple
number of clients: 4
number of threads: 1
duration: 60 s
number of transactions actually processed: 34324
tps = 571.932873 (including connections establishing)
tps = 572.082084 (excluding connections establishing)
```

Run pgfouine on today's log file to make sure it works.

*note: The command starts with pgfouine.php not just pgfouine.

```
-bash-4.1$ pgfouine.php -logtype stderr -file /var/lib/pgsql/9.1/data/pg_log/postgresql-wed.log
```

```
##### Overall statistics #####
```

```
Number of unique normalized queries: 1
Number of queries: 1
Total query duration: 0.0s
```

```
##### Queries by type #####
```

```
SELECT: 1 100.0%
```

```
##### Queries that took up the most time (N) #####
```

```
1) 0.0s - 1 - select count(*) from pgbench_branches
--
```

```
##### Slowest queries #####
```

```
1) 0.00 s - select count(*) from pgbench_branches
--
```

```
##### Most frequent queries (N) #####
```

```
1) 1 - 0.0s - select count(*) from pgbench_branches
--
```

```
##### Slowest queries (N) #####
```

```
1) 0.00 - 1 - select count(*) from pgbench_branches
--
```

Posted by [drolicheck](#) at Jul 17, 2013.

Edit postgresql.conf log_filename value so we don't overwrite log files every week.

```
-bash-4.1$ diff postgresql.conf.130723 postgresql.conf
296c296
```

```
< log_filename = 'postgresql-%a.log' # log file name pattern,  
---  
> log_filename = 'postgresql-%Y%m%d.log'      # log file name pattern,
```

Example of old file name format compared to new format.

Old format:

postgresql-Fri.log

New format:

postgresql-20130723.log

Posted by [drolicheck](#) at Jul 23, 2013.

Recently re-edited and to-be-reviewed navigation files

This page last changed on Mar 05, 2009 by [linda](#).

[illegible]

Recently re-edited and to-be-reviewed navigation files*

This page last changed on Jul 30, 2009 by [dana](#).

				These cannot be sorted! Do a ctrl F to find what you are looking for						
Vehicle	Year	date	Date	Chief Scientist	Dive #	Complete or Reviewed	Status?	By	To McCann for re-load	To Lonny for VARS re-merge
DOCR	2009				D0001-D0041	22-Jun-2009	bad navigation data for these dives	linda	na	na
Ventana	2009	106	04/16/2009	Rock	V3351, 3352	28-Apr-09	cannot get rid of several large spikes	linda	na	na
Tiburon	2004	252	09/09/2009	Stakes	T741	10-Mar-09	ok	linda	x	x
Tiburon	2004	251	09/07/2009	Stakes	T740-741	10-Mar-09	ok	linda	x	x
Tiburon	2004	250	09/06/2009	Stakes	T739-740	10-Mar-09	ok	linda	x	x
Tiburon	2004	249	09/05/2009	Stakes	T738-739	10-Mar-09	ok	linda	x	x
Tiburon	2004	248	09/04/2009	Stakes	T737-738	10-Mar-09	ok	linda	x	x
Tiburon	2004	247	09/03/2009	Stakes	T737	10-Mar-09	ok	linda	x	x
Ventana	2009	042	02/11/2009	Robison	V3335	10-Mar-09	several large spikes cannot be smoothed	linda	x	x

Ventana	2009041	02/10/2008	Robison	V3334	10-Mar-09	2 large spikes cannot be smoothed	linda	x	x
Tiburon	2004244	08/31/2008	Stakes	T735	09-Mar-09	ok	linda	x	x
Tiburon	2004245	09/01/2008	Stakes	T735-736	09-Mar-09	ok	linda	x	x
Tiburon	2004246	09/02/2008	Stakes	T736	09-Mar-09	ok	linda	x	x
Ventana	2008354	12/19/2008	Wankel	V3317	09-Mar-09	ok	dana	x	x
Ventana	2008353	12/18/2008	Massion	V3316	09-Mar-09	ok	dana	x	x
Ventana	2008346	12/11/2008	Hester	V3311	06-Mar-09	not needed / GE	dana	na	na
Ventana	2008345	12/10/2008	Walz	V3310	06-Mar-09	not needed / GE	dana	na	na
Ventana	2008344	12/9/2008	Walz	V3309	09-Mar-09	ok	dana	x	x
Ventana	2008340	12/5/2008	Ussler	V3307, V3308	06-Mar-09	ok	dana	x	x
Ventana	2008337	12/2/2008	McClain	V3306	09-Mar-09	ok	dana	x	x
Ventana	2008336	12/1/2008	Vrijenhoek	V3304, V3305	09-Mar-09	not needed / GE	linda	na	na
Ventana	2008331	11/26/2008	Johnson	V3303	06-Mar-09	ok	dana	x	x
Ventana	2008330	11/25/2008	Baull	V3302	06-Mar-09	ok	dana	x	x
Ventana	2008326	11/21/2008	Whaling	V3300, V3301	09-Mar-09	not needed	linda	na	na
Ventana	2008324	11/19/2008	Brewer	V3299	09-Mar-09	ok	dana	x	x
Ventana	2008323	11/18/2008	Brewer	V3298	09-Mar-09	ok	dana	x	x
Ventana	2008318	11/13/2008	Bawe	3297	11-Mar-09	ok	dana		
Ventana	2008317	11/12/2008	McGill	3296	11-Mar-09	ok	dana		

Ventana	2008316	11/11/2008	Brekke	3295	11-Mar-09	ok	dana		
Ventana	2008315	11/10/2008	Raybould	3294	11-Mar-09	one spike to 15 km/hr	dana		
Ventana	2008302	10/28/2008	Massion	3288, 3289	11-Mar-09	ok	dana		
Ventana	2008301	10/27/2008	Smith	3287	11-Mar-09	ok	dana		
Ventana	2008289	10/15/2008	Whaling	3281	29-Apr-09	ok	dana		
Ventana	2008288	10/14/2008	Smith	3280	29-Apr-09	ok	dana		
Ventana	2008283	10/9/2008	Widmer	3279	29-Apr-09	ok	dana		
Ventana	2008281	10/7/2008	Girguis	3278	29-Apr-09	ok	dana		
Ventana	2008275	10/1/2008	Walz	3277	16-Mar-09	test tank	dana		
Ventana	2008273	9/29/2008	Smith	3276	16-Mar-09	ok	dana		
Ventana	2008270	9/26/2008	Rock	3275	16-Mar-09	ok	dana		
Ventana	2008269	9/25/2008	Raymond	3273, 3274	09-Apr-09	ok	dana		
Ventana	2008268	9/24/2008	Massion	3272	09-Apr-09	ok	dana		
Ventana	2008249	9/5/2008	Raymond/Vrijenhoek	3266	09-Apr-09	ok	dana		
Ventana	2008242	8/29/2008	Brewer	3263	09-Apr-09	ok	dana		
Ventana	2008241	8/28/2008	Brewer	3262	09-Apr-09	ok	dana		
Ventana	2008240	8/27/2008	Brewer	3261	09-Apr-09	ok	dana		
Ventana	2008239	8/26/2008	Peltzer	3258, 3259	16-Apr-09	ok	dana		
Ventana	2008238	8/25/2008	Brewer	3257	16-Apr-09	ok	dana		
Ventana	2008234	8/21/2008	Whaling	3253, 3254, 3255, 3256	16-Apr-09	ok	dana		

Ventana	2008233	8/20/2008	Raybould	3252	16-Apr-09	ok	dana		
Ventana	2008231	8/18/2008	McGill	3251	16-Apr-09	ok	dana		
Ventana	2008220	8/7/2008	Rock	3245	20-Apr-09	large spike at beginning, questionable nav data for last 1/3 of dive	dana		
Ventana	2008219	8/6/2008	Kelley	3244	20-Apr-09	ok	dana		
Ventana	2008218	8/5/2008	Paull	3242, 3243	20-Apr-09	ok	dana		
Ventana	2008213	7/31/2008	Rock	3241	20-Apr-09	ok	dana		
Ventana	2008212	7/30/2008	Rock	3239	20-Apr-09	ok	dana		
Ventana	2008205	7/23/2008	Peltzer	3236, 3237	20-Apr-09	ok	dana		
Ventana	2008204	7/22/2008	Brewer	3235	20-Apr-09	ok	dana		
Ventana	2008198	7/16/2008	Whaling	3231, 3232	20-Apr-09	ok	dana		
Ventana	2008197	7/15/2008	McGill	3230	20-Apr-09	ok	dana		
Ventana	2008185	7/3/2008	Widmer	3228	20-Apr-09	30km/h spike, 15km/h spikes, ugly nav	dana		
Ventana	2008182	6/30/2008	Whaling	3225, 3226	20-Apr-09	ok	dana		
Ventana	2008172	6/20/2008	Augenstein	3224	20-Apr-09	ok	dana		
Ventana	2008170	6/18/2008	Kirkwood	3223	20-Apr-09	ok	dana		
Ventana	2008169	6/17/2008	Kirkwood	3221, 3222	23-Apr-09	bad data in 2nd dive	dana		
Ventana	2008165	6/13/2008	Raymond	3218, 3219, 3220	23-Apr-09	ok	dana		

				(no video)					
Ventana	2008164	6/12/2008	Whaling	3215, 3216, 3217	23-Apr-09	ok	dana		
Ventana	2008162	6/10/2008	Murthy	3214	23-Apr-09	ok	dana		
Ventana	2008158	6/6/2008	Barry	3211, 3212, 3213					
Ventana	2008157	6/5/2008	Massion	3210	23-Apr-09	edited nav had "hiccups" sent to LAK	dana		
Ventana	2008151	5/30/2008	Raymond	3207	23-Apr-09	all data pre-selected upon opening	dana		
Ventana	2008150	5/29/2008	Brewer	3206	23-Apr-09	ok, 2 spikes to 16 km/h	dana		
Ventana	2008136	5/15/2008	Vrijenhoek	3202	27-Apr-09	ok	dana		
Ventana	2008133	5/12/2008	Brewer	3201	27-Apr-09	ok	dana		
Ventana	2008126	5/5/2008	Rock	3200	27-Apr-09	ok	dana		
Ventana	2008120	4/29/2008	Smith	3199	27-Apr-09	ok	dana		
Ventana	2008109	4/18/2008	McClain	3193	27-Apr-09	long spikes toward end of dive	dana		
Ventana	2008108	4/17/2008	Smith	3192	27-Apr-09	ok	dana		
Ventana	2008101	4/10/2008	Whaling	3191	27-Apr-09	ok	dana		
Ventana	2008100	4/9/2008	Widmer	3190	27-Apr-09	ok	dana		
Ventana	2008095	4/4/2008	Rock	3189	27-Apr-09	ok	dana		
Ventana	2008093	4/2/2008	Whaling	3187, 3188	27-Apr-09	1st dive ok; 2nd dive: 2	dana		

						spikes near end			
Ventana	2008092	4/1/2008	Whaling	3186	27- Apr-09	ok	dana		
Ventana	2008091	3/31/2008	Whaling	3183, 3184, 3185	27- Apr-09	strange depth data: no usable navedit	dana		
Ventana	2008060	2/29/2008	Dawe	3179	28- Apr-09	ok	dana		
Ventana	2008059	2/28/2008	Dawe	3178	28- Apr-09	ok	dana		
Ventana	2008057	2/26/2008	Raybould	3177	28- Apr-09	"Duplicate or reverse order time stamps detected"	dana		
Ventana	2008050	2/19/2008	McGill	3176	28- Apr-09	ok	dana		
Ventana	2008045	2/14/2008	Kuhnz	3173, 3174, 3175	28- Apr-09	ok	dana		
Ventana	2008044	2/13/2008	Rock	3171	28- Apr-09	ok	dana		
Ventana	2008043	2/12/2008	Bird	3170	28- Apr-09	ok	dana		
Ventana	2008039	2/8/2008	Kuhnz	3169	28- Apr-09	ok	dana		
Ventana	2008031	1/31/2008	Kuhnz	3167	28- Apr-09	ok	dana		
Ventana	2008030	1/30/2008	Kuhnz	3165, 3166	28- Apr-09	ok	dana		
Ventana	2008029	1/29/2008	Kuhnz	3164	29- Apr-09	ok	dana		
Ventana	2008028	1/28/2008	Kuhnz	3163	29- Apr-09	ok	dana		
Ventana	2008024	1/24/2008	Kuhnz	3162	29- Apr-09	ok	dana		
Ventana	2008023	1/23/2008	Whaling	3160, 3161	29- Apr-09	ok	dana		
Ventana	2008011	1/11/2008	Widmer	3154	29- Apr-09	ok	dana		

Ventana	2008010	1/10/2008	Whaling	3151, 3152, 3153	29- Apr-09	ok	dana		
Ventana	2008009	1/9/2008	Vrijenhoek	3150	29- Apr-09	ok	dana		
Ventana	2008008	1/8/2008	Whaling	3148, 3149	29- Apr-09	ok	dana		
Ventana	2008007	1/7/2008	Kuhnz	3147	29- Apr-09	ok	dana		
Tiburon	2007354	12/20/2007	Vrijenhoek	1163					
Tiburon	2007352	12/19/2007	Vrijenhoek	1161					
Tiburon	2007353	12/19/2007	Vrijenhoek	1162					
Tiburon	2007352	12/18/2007	Vrijenhoek	1160, 1161					
Ventana	2007347	12/13/2007	Smith	3140, 3141	29- Apr-09	ok	dana		
Ventana	2007346	12/12/2007	Whaling	3139	29- Apr-09	ok	dana		
Ventana	2007345	12/11/2007	Whaling	3137, 3138	29- Apr-09	ok	dana		
Tiburon	2007344	12/10/2007	Chaffey	1159					
Tiburon	2007343	12/9/2007	Chaffey	1158					
Ventana	2007340	12/6/2007	Kuhnz	3136	29- Apr-09	questionable data, no re- edit	dana		
Ventana	2007334	11/30/2007	Kuhnz	3134, 3135	29- Apr-09	questionable data, no re- edit	dana		
Ventana	2007332	11/28/2007	Whaling	3133	29- Apr-09	ok	dana		
Ventana	2007330	11/26/2007	Smith	3132	29- Apr-09	ok	dana		
Tiburon	2007324	11/19/2007	Barry	1153					
Ventana	2007312	11/8/2007	Whaling	3128, 3129	30- Apr-09	ok	dana		
Ventana	2007311	11/7/2007	Smith	3127	30- Apr-09	ok	dana		
Ventana	2007306	11/2/2007	Peltzer	3126	30- Apr-09	ok	dana		

Ventana	2007303	10/30/2007	Rock	3125	30-Apr-09	ok	dana		
Ventana	2007302	10/29/2007	McGill	3124	30-Apr-09	questionable data, no re-edit	dana		
Ventana	2007297	10/24/2007	Brewer	3123	30-June-09	ok, flat nav for last third of dive	dana		
Tiburon	2007296	10/23/2007	Barry	1145					
Ventana	2007295	10/22/2007	Paul	3118, 3119, 3120, 3121	30-June-09	ok	dana		
Tiburon	2007295	10/22/2007	Barry	1144					
Ventana	2007292	10/19/2007	Paul	3113, 3114, 3115, 3116, 3117	06-May-09	ok	dana		
Ventana	2007291	10/18/2007	Paul	3107, 3108, 3109, 3110, 3111, 3112	06-May-09	questionable data, did not re-edit	dana		
Ventana	2007289	10/16/2007	Paul	3102, 3103, 3104, 3105, 3106	26-June-09	ok	dana		
Ventana	2007288	10/15/2007	Paul	3096, 3097, 3098, 3099, 3100, 3101	26-June-09	ok	dana		
Ventana	2007277	10/4/2007	Conway	3090	26-June-09	ok	dana		
Ventana	2007271	9/28/2007	Whaling	3088	26-June-09	ok	dana		
Tiburon	2007266	9/23/2007	Smith	1143					
Tiburon	2007265	9/22/2007	Smith	1142					
Tiburon	2007264	9/21/2007	Smith	1141					
Tiburon	2007263	9/20/2007	Smith	1140					

Tiburon	2007262	9/19/2007	Smith	1139					
Tiburon	2007261	9/18/2007	Smith	1138					
Ventana	2007256	9/13/2007	Vrijenhoek	3077	26-June-09	ok	dana		
Ventana	2007255	9/12/2007	Whaling	3075, 3076	26-June-09	ok	dana		
Ventana	2007253	9/10/2007	McGill	3074	26-June-09	ok	dana		
Tiburon	2007253	9/10/2007	Paull	1136, 1137					
Tiburon	2007252	9/9/2007	Paull	1133, 1134, 1135					
Tiburon	2007251	9/8/2007	Paull	1129, 1130, 1131, 1132					
Tiburon	2007250	9/7/2007	Paull	1128					
Ventana	2007249	9/6/2007	Johnson	3072, 3073	24-Jun-09	ok	dana		
Tiburon	2007249	9/6/2007	Paull	1127					
Tiburon	2007248	9/5/2007	Paull	1126					
Ventana	2007247	9/4/2007	Johnson	3070, 3071	24-Jun-09	ok	dana		
Tiburon	2007243	8/31/2007	Barry	1125					
Tiburon	2007242	8/30/2007	Barry	1124					
Tiburon	2007241	8/29/2007	Barry	1123					
Ventana	2007240	8/28/2007	Dawe	3069	24-Jun-09	ok	dana		
Tiburon	2007240	8/28/2007	Barry	1122					
Tiburon	2007239	8/27/2007	Barry	1121					
Ventana	2007229	8/17/2007	Whaling	3065	24-Jun-09	ok	dana		
Tiburon	2007229	8/17/2007	Vrijenhoek	1120					
Ventana	2007228	8/16/2007	Rock	3064	24-Jun-09	ok	dana		
Tiburon	2007228	8/16/2007	Vrijenhoek	1119					
Ventana	2007227	8/15/2007	Brewer	3063	24-Jun-09	ok	dana		

Tiburon	2007227	8/15/2007	Vrijenhoek	1117, 1118					
Ventana	2007226	8/14/2007	Brewer	3062	24-Jun-09	ok	dana		
Ventana	2007214	8/2/2007	Whaling	3056, 3057, 3058	24-Jun-09	ok	dana		
Ventana	2007212	7/31/2007	Johnson	3053, 3054, 3055	24-Jun-09	ok	dana		
Ventana	2007190	7/9/2007	Whaling	3048, 3049, 3050	23-Jun-09	ok	dana		
Ventana	2007180	6/28/2007	Paull	3041, 3042, 3043, 3044, 3045, 3046, 3047	23-Jun-09	no file	dana		
Ventana	2007179	6/27/2007	Paull	3034, 3035, 3036, 3037, 3038, 3039, 3040	23-Jun-09	ok	dana		
Ventana	2007178	6/26/2007	Paull	3029, 3030, 3031, 3032, 3033	23-Jun-09	ok	dana		
Tiburon	2007175	6/24/2007	Clague	1103/1104					
Ventana	2007173	6/22/2007	Sherman	3028	23-Jun-09	ok	dana		
Ventana	2007172	6/21/2007	Sherman	3027	23-Jun-09	ok	dana		
Tiburon	2007171	6/20/2007	Clague	1102					
Tiburon	2007170	6/19/2007	Clague	1101					
Ventana	2007169	6/18/2007	Whaling	3024	23-Jun-09	ok	dana		
Tiburon	2007169	6/18/2007	Clague	1100					
Ventana	2007166	6/15/2007	Rock	3023	23-Jun-09	ok	dana		
Ventana	2007165	6/14/2007	Ramirez	3022	23-Jun-09	ok	dana		

Ventana	2007164	6/13/2007	Whaling	3019	23-Jun-09	ok	dana		
Tiburon	2007161	6/10/2007	Smith	1099					
Tiburon	2007160	6/9/2007	Smith	1098					
Tiburon	2007159	6/8/2007	Smith	1096					
Ventana	2007158	6/7/2007	Dawe	3016	18-Jun-09	ok	dana		
Tiburon	2007158	6/7/2007	Smith	1095					
Tiburon	2007156	6/5/2007	Smith	1094					
Ventana	2007155	6/4/2007	Jones	3014	23-Jun-09	re-edited, flat nav from 18:41 to end	dana		
Ventana	2007151	5/31/2007	Peltzer	3013	23-Jun-09	no reedit, bad nav	dana		
Ventana	2007150	5/30/2007	Barry	3012	23-Jun-09	re-edited, flat nav from 17:57 to end	dana		
Ventana	2007149	5/29/2007	Smith	3011	18-Jun-09	ok	dana		
Ventana	2007145	5/25/2007	Rock	3009	07-July-09	ok	dana		
Ventana	2007142	5/22/2007	Brewer	3008	01-July-09	ok, flat nav from 21:54 to end	dana		
Ventana	2007131	5/11/2007	Paull	3003, 3004	18-Jun-09	ok	dana		
Ventana	2007129	5/9/2007	Paull	3001, 3002	18-Jun-09	questionable nav, no re-edit	dana		
Ventana	2007128	5/8/2007	Paull	2999, 3000	18-Jun-09	ok	dana		
Ventana	2007124	5/4/2007	Whaling	2997	18-Jun-09	ok	dana		
Ventana	2007123	5/3/2007	Rock	2996	18-Jun-09	ok	dana		

Ventana	2007122	5/2/2007	Johnson	2994/2995	18-Jun-09	ok	dana		
Tiburon	2007119	4/29/2007	Paull	1093					
Tiburon	2007118	4/28/2007	Paull	1092					
Tiburon	2007117	4/27/2007	Paull	1091					
Ventana	2007114	4/24/2007	Brewer	2991	18-Jun-09	ok	dana		
Tiburon	2007114	4/24/2007	Barry	1090					
Ventana	2007113	4/23/2007	Brewer	2990	18-Jun-09	ok	dana		
Tiburon	2007113	4/23/2007	Barry	1089					
Ventana	2007109	4/19/2007	Whaling	2989	18-Jun-09	ok	dana		
Tiburon	2007084	3/25/2007	Paull/ Barry	1082					
Tiburon	2007083	3/24/2007	Paull/ Barry	1081					
Ventana	2007081	3/22/2007	Dawe	2988	17-June-09	ok	dana		
Ventana	2007079	3/20/2007	Dawe	2987	17-June-09	ok	dana		
Ventana	2007078	3/19/2007	Dawe	2986	17-June-09	ok	dana		
Ventana	2007076	3/17/2007	Dawe	2985	17-June-09	questionable nav, no re-edit	dana		
Ventana	2007075	3/16/2007	Dawe	2984	17-June-09	ok	dana		
Ventana	2007059	2/28/2007	Whaling	2979	17-June-09	ok	dana		
Ventana	2007058	2/27/2007	Whaling	2976	17-June-09	questionable nav, no re-edit	dana		
Ventana	2007045	2/14/2007	Johnson/ Barry	2970, 2971, 2972	17-June-09	questionable nav, no re-edit	dana		
Ventana	2007043	2/12/2007	Paull	2966, 2967, 2968, 2969	17-June-09	ok	dana		
Ventana	2007039	2/8/2007	Paull	2960, 2961, 2962,	9-June-09	ok	dana		

				2963, 2964, 2965					
Tiburon	2007035	2/4/2007	Smith	1080					
Tiburon	2007034	2/3/2007	Smith	1079					
Tiburon	2007033	2/2/2007	Smith	1078					
Ventana	2007032	2/1/2007	Vrijenhoek	2958	9- June-09	ok	dana		
Tiburon	2007032	2/1/2007	Smith	1077					
Ventana	2007031	1/31/2007	McGill	2957	9- June-09	ok	dana		
Tiburon	2007031	1/31/2007	Smith	1076					
Ventana	2007023	1/23/2007	Kelley	2951	9- June-09	ok	dana		
Ventana	2007017	1/17/2007	Jones	2948	9- June-09	ok	dana		
Ventana	2007016	1/16/2007	Johnson	2945, 2946, 2947	9- June-09	questionable nav, no re-edit	dana		
Ventana	2007012	1/12/2007	Rock	2944	9- June-09	ok	dana		
Tiburon	2007012	1/12/2007	Vrijenhoek	1071, 1072					
Tiburon	2007011	1/11/2007	Vrijenhoek	1070					
Ventana	2007010	1/10/2007	Johnson	2941, 2942, 2943	9- June-09	questionable nav, no re-edit	dana		
Tiburon	2007010	1/10/2007	Vrijenhoek	1069					
Ventana	2007005	1/5/2007	Paull	2940	9- June-09	no file	dana		
Ventana	2007002	1/2/2007	Smith	2938	9- June-09	ok	dana		
Ventana	2006352	12/18/2006	Smith	2937	8- June-09	no re- edit, bad nav	dana		
Tiburon	2006348	12/14/2006	Smith	1068					
Tiburon	2006347	12/13/2006	Smith	1067					
Tiburon	2006346	12/12/2006	Smith	1066					
Ventana	2006340	12/6/2006	Whaling	2932	8- June-09	ok	dana		

Ventana	2006339	12/5/2006	Smith	2931	8-June-09	ok	dana		
Ventana	2006338	12/4/2006	Jensen	2930	8-June-09	ok	dana		
Ventana	2006334	11/30/2006	Smith	2929	8-June-09	ok	dana		
Ventana	2006332	11/28/2006	Girguis	2928	8-June-09	ok	dana		
Ventana	2006331	11/27/2006	Girguis	2926, 2927	8-June-09	ok	dana		
Ventana	2006325	11/21/2006	Brewer	2925	4-June-09	ok	dana		
Ventana	2006324	11/20/2006	Brewer	2924	4-June-09	ok	dana		
Tiburon	2006322	11/18/2006	Barry	1065					
Tiburon	2006321	11/17/2006	Barry	1064					
Ventana	2006320	11/16/2006	Johnson	2923	4-June-09	ok	dana		
Tiburon	2006320	11/16/2006	Barry	1063					
Tiburon	2006319	11/15/2006	Barry	1062					
Tiburon	2006318	11/14/2006	Barry	1061					
Tiburon	2006310	11/6/2006	Paull	1060					
Tiburon	2006309	11/5/2006	Paull	1059					
Tiburon	2006308	11/4/2006	Paull	1058					
Tiburon	2006307	11/3/2006	Paull	1057					
Ventana	2006306	11/2/2006	Whaling	2919, 2920	4-June-09	no re-edit, questionable nav	dana		
Tiburon	2006305	11/1/2006	Paull	1053,1054,1055,1056					
Tiburon	2006304	10/31/2006	Baull	1051,1052					
Tiburon	2006299	10/26/2006	Vrijenhoek	1050					
Tiburon	2006298	10/25/2006	Vrijenhoek	1049					
Tiburon	2006297	10/24/2006	Vrijenhoek	1048					
Tiburon	2006293	10/20/2006	Baull	1047					
Tiburon	2006292	10/19/2006	Baull	1046	30-July-09	ok	dana		

Tiburon	2006291	10/18/2006	Baull	1045	30-July-09	ok	dana		
Tiburon	2006290	10/17/2006	Baull	1044					
Ventana	2006285	10/12/2006	Whaling	2908, 2909	4-June-09	ok	dana		
Ventana	2006284	10/11/2006	Brewer	2907	4-June-09	ok	dana		
Ventana	2006282	10/9/2006	Brewer	2906	4-June-09	ok	dana		
Ventana	2006278	10/5/2006	Whaling	2903, 2904, 2905	4-June-09	ok	dana		
Ventana	2006277	10/4/2006	Whaling	2900, 2901, 2902	4-June-09	ok	dana		
Ventana	2006276	10/3/2006	Whaling	2898, 2899	4-June-09	ok	dana		
Ventana	2006275	10/2/2006	McGill	2897	4-June-09	ok	dana		
Ventana	2006271	9/28/2006	Widder	2895, 2896	3-June-09	ok	dana		
Ventana	2006269	9/26/2006	Widder	2893, 2894	3-June-09	ok	dana		
Ventana	2006268	9/25/2006	Whaling	2890, 2891, 2892	3-June-09	ok	dana		
Tiburon	2006265	9/22/2006	Grech	1034	30-July-09	ok	dana		
Tiburon	2006264	9/21/2006	Grech	1032, 1033	30-July-09	ok	dana		
Ventana	2006263	9/20/2006	Girguis	2888, 2889	3-June-09	ok	dana		
Tiburon	2006263	9/20/2006	Grech	1031	30-July-09	ok	dana		
Ventana	2006262	9/19/2006	Johnson	2887	3-June-09	ok	dana		
Tiburon	2006262	9/19/2006	Grech	1030	30-July-09	ok	dana		
Ventana	2006261	9/18/2006	Whaling	2884, 2885, 2886	3-June-09	questionable, no re-edit	dana		
Tiburon	2006261	9/18/2006	Grech	1029	30-July-09	ok	dana		

Ventana	2006254	9/11/2006	Barry	2879, 2880	3- June-09	ok	dana		
Ventana	2006250	9/7/2006	Gary Wilson	2878	3- June-09	ok	dana		
Ventana	2006249	9/6/2006	Whaling	2876, 2877	3- June-09	questionable nav, no re-edit	dana		
Ventana	2006248	9/5/2006	Vrijenhoek	2874, 2875	3- June-09	questionable nav, no re-edit	dana		
Tiburon	2006229	8/17/2006	Brewer	1022	27- July-09	ok	dana		
Tiburon	2006228	8/16/2006	Brewer	1020, 1021	27- July-09	ok	dana		
Ventana	2006227	8/15/2006	Rock	2873	3- June-09	questionable nav, no re-edit	dana		
Tiburon	2006227	8/15/2006	Brewer	1019	27- July-09	ok	dana		
Tiburon	2006225	8/13/2006	Brewer	1018	27- July-09	ok	dana		
Tiburon	2006224	8/12/2006	Brewer	1017	27- July-09	ok	dana		
Tiburon	2006223	8/11/2006	Brewer	1016	27- July-09	ok	dana		
Tiburon	2006217	8/5/2006	Clague	1014	27- July-09	ok	dana		
Tiburon	2006216	8/4/2006	Clague	1013	27- July-09	ok	dana		
Ventana	2006215	8/3/2006	Whaling	2864, 2865	3- June-09	questionable nav, no re-edit	dana		
Tiburon	2006215	8/3/2006	Clague	1012	27- July-09	ok	dana		
Tiburon	2006214	8/2/2006	Clague	1011	27- July-09	ok	dana		
Tiburon	2006213	8/1/2006	Clague	1010	27- July-09	ok	dana		
Ventana	2006212	7/31/2006	Johnson	2861,2862	23- June-09	questionable nav, no re-edit	dana		
Tiburon	2006212	7/31/2006	Clague	1009	27- July-09	ok	dana		

Tiburon	2006211	7/30/2006	Clague	1008	27-July-09	ok	dana		
Tiburon	2006210	7/29/2006	Clague	1007	27-July-09	ok	dana		
Ventana	2006208	7/27/2006	Rock	2859	3-June-09	questionable nav, no re-edit	dana		
Ventana	2006206	7/25/2006	Vrijenhoek	2857, 2858	3-June-09	ok	dana		
Ventana	2006201	7/20/2006	Whaling	2856	3-June-09	ok	dana		
Ventana	2006199	7/18/2006	McGill	2855	3-June-09	ok	dana		
Ventana	2006198	7/17/2006	Johnson	2853, 2854	3-June-09	questionable nav, no re-edit	dana		
Ventana	2006194	7/13/2006	Whaling	2852	3-June-09	ok	dana		
Tiburon	2006193	7/12/2006	Paull	1006	23-July-09	ok	dana		
Ventana	2006192	7/11/2006	McGill	2851	3-June-09	ok	dana		
Tiburon	2006192	7/11/2006	Paull	1005	23-July-09	ok	dana		
Tiburon	2006191	7/10/2006	Paull	1004	23-July-09	ok	dana		
Tiburon	2006190	7/9/2006	Paull	1003	23-July-09	ok	dana		
Tiburon	2006189	7/8/2006	Paull	1002	23-July-09	ok	dana		
Ventana	2007178	6/26/2006	Paull	3029, 3030, 3031, 3032, 3033	1-June-09	ok	dana		
Tiburon	2006176	6/25/2006	Rock	1001	23-July-09	ok	dana		
Ventana	2006173	6/22/2006	Vrijenhoek	2841	1-June-09	questionable nav, no re-edit	dana		
Ventana	2006166	6/15/2006	Madin	2840	1-June-09	questionable nav, no re-edit	dana		

Ventana	2006165	6/14/2006	Madin	2839	1- June-09	questionable nav, no re-edit	dana		
Ventana	2006163	6/12/2006	Madin	2837	no file		dana		
Ventana	2006152	6/1/2006	Whaling	2830, 2831, 2832, 2833	1- June-09	questionable nav, no re-edit	dana		
Ventana	2006151	5/31/2006	McGill	2829	1- June-09	ok	dana		
Ventana	2006150	5/30/2006	Paull	2826, 2827, 2828	1- June-09	ok	dana		
Tiburon	2006146	5/26/2006	Rock	994	23- July-09	ok	dana		
Tiburon	2006145	5/25/2006	Vrijenhoek	992, 993	23- July-09	ok	dana		
Ventana	2006144	5/24/2006	Widder	2825	1- June-09	ok	dana		
Tiburon	2006144	5/24/2006	Vrijenhoek	991	23- July-09	ok	dana		
Ventana	2006143	5/23/2006	Whaling	2824	1- June-09	ok	dana		
Tiburon	2006143	5/23/2006	Vrijenhoek	990	23- July-09	ok; flatlines from 21:30 to end	dana		
Ventana	2006142	5/22/2006	Widder	2822, 2823	1- Jun-09	ok	dana		
Ventana	2006138	5/18/2006	Widder	2820, 2821	12- May-09	ok	dana		
Ventana	2006137	5/17/2006	Whaling	2817, 2818, 2819	12- May-09	questionable nav, no re-edit	dana		
Ventana	2006136	5/16/2006	Widder	2815	12- May-09	ok	dana		
Ventana	2006135	5/15/2006	Rock	2814	12- May-09	ok	dana		
Ventana	2006131	5/11/2006	Whaling	2813	12- May-09	ok	dana		
Ventana	2006123	5/3/2006	Brewer	2810	12- May-09	ok	dana		
Ventana	2006121	5/1/2006	Peltzer	2809	12- May-09	ok	dana		

Tiburon	2006120	4/30/2006	Paull	977, 978, 979	21- July-09	ok, a little spiky (16 km/ h)	dana		
Tiburon	2006119	4/29/2006	Paull	975, 976	21- July-09	ok, a little spiky (16 km/ h)	dana		
Tiburon	2006118	4/28/2006	Paull	972, 973, 974	21- July-09	ok	dana		
Ventana	2006117	4/27/2006	Rock	2808	12- May-09	questionable nav, no re-edit	dana		
Tiburon	2006117	4/27/2006	Paull	971	21- July-09	ok	dana		
Tiburon	2006110	4/20/2006	Brewer	970	21- July-09	ok	dana		
Tiburon	2006109	4/19/2006	Brewer	968, 969	21- July-09	ok	dana		
Tiburon	2006108	4/18/2006	Brewer	966, 967	21- July-09	ok	dana		
Tiburon	2006085	3/26/2006	Paull/ Barry	957, 958	16- July-09	ok, but spiky	dana		
Tiburon	2006084	3/25/2006	Paull/ Barry	956	16- July-09	ok, but spiky	dana		
Tiburon	2006083	3/24/2006	Paull/ Barry	955	16- July-09	ok, but spiky	dana		
Tiburon	2006082	3/23/2006	Paull/ Barry	954	16- July-09	ok	dana		
Tiburon	2006081	3/22/2006	Paull/ Barry	953	16- July-09	ok	dana		
Ventana	2006067	3/8/2006	Jones	2800	12- May-09	ok	dana		
Ventana	2006066	3/7/2006	Johnson	2799	12- May-09	ok	dana		
Ventana	2006065	3/6/2006	Keaten	2797	12- May-09	ok	dana		
Ventana	2006044	2/13/2006	Widmer	2791	12- May-09				
Ventana	2006040	2/9/2006	Whaling	2789, 2790	12- May-09	ok	dana		

Ventana	2006039	2/8/2006	Brewer	2788	11-May-09	ok	dana		
Ventana	2006038	2/7/2006	Brewer	2787	11-May-09	ok	dana		
Ventana	2006037	2/6/2006	Brewer	2786	11-May-09	ok	dana		
Tiburon	2006034	2/3/2006	DeVogelaes	950	04-Feb-09	ok	linda	04-Feb-09	x
Tiburon	2006033	2/2/2006	DeVogelaes, BBC	948, 949	04-Feb-09	ok	linda	04-Feb-09	x
Ventana	2006032	2/1/2006	Fitzwater/Plant	2785	11-May-09	ok	dana		
Tiburon	2006032	2/1/2006	DeVogelaes	947	04-Feb-09	ok	linda	04-Feb-09	x
Tiburon	2006031	1/31/2006	DeVogelaes	946	15-July-09	ok	dana		
Ventana	2006030	1/30/2006	McGill	2784	11-May-09	questionable nav, no re-edit	dana		
Tiburon	2006030	1/30/2006	DeVogelaes	945	15-July-09	ok	dana		
Tiburon	2006029	1/29/2006	Allen-BBC	944	15-July-09	ok	dana		
Tiburon	2006028	1/28/2006	DeVogelaes	943	15-July-09	ok	dana		
Tiburon	2006027	1/27/2006	DeVogelaes	940, 941	15-July-09	ok	dana		
Ventana	2006026	1/26/2006	Rock	2783	11-May-09	large spikes, no re-edit	dana		
Tiburon	2006026	1/26/2006	DeVogelaes	939	13-July-09	no file	dana		
Ventana	2006025	1/25/2006	Jones	2782	11-May-09	ok	dana		
Tiburon	2006021	1/21/2006	Barry	938	13-July-09	ok	dana		
Tiburon	2006020	1/20/2006	Barry	937	13-July-09	bad data, no reedit	dana		
Tiburon	2006019	1/19/2006	Barry	936	13-July-09	bad data,	dana		

						no reedit			
Tiburon	2006018	1/18/2006	Barry	934, 935	13- July-09	first dive bad nav; 2nd ok	dana		
Ventana	2006017	1/17/2006	Paull	2780	11- May-09	ok	dana		
Ventana	2006012	1/12/2006	Whaling	2778	11- May-09	2nd dive? bad data, no re- edit	dana		
Tiburon	2006006	1/6/2006	Vrijenhoek	933	13- July-09	ok	dana		
Tiburon	2006005	1/5/2006	Vrijenhoek	932	13- July-09	ok	dana		
Ventana	2006004	1/4/2006	Barry	2774, 2775	11- May-09	most points pre- selected upon opening, no re- edit	dana		
Tiburon	2006006	1/4/2006	Vrijenhoek	931					

ROVGISUpgradeMeeting03Oct2007

This page last changed on Oct 03, 2007 by [mccann](#).

Attendees: Mike McCann, Jenny Paduan, Jim Barry, Dave Clague

Narrowing the options

We will pursue option A, the first bullet in the proposal. Namely:

Upgrade to arcGIS via development of visual basic scripts to rewrite ArcNAV.
Although development of .avenue scripts for the original ArcNAV extension for Arcview software was relatively straightforward, a similar approach using Visual Basic for ArcGIS is thought to be considerably more complex owing to the structure of ESRI's ArcGIS software.

The approach that has already proven quite tractable is to write an ArcGIS extension in C# using VisualStudio. The complexity is in understanding the ArcGIS object model - which exists in any of the languages available for extending ArcGIS. Programming the extension at this level will give the greatest amount of flexibility in adding the features that are required.

(Even though this approach is the apparent way to go we do want to double check that the Tracking Analyst option actually does require a connection to shore, and a server in Redlands, CA.)

Features in the ArcView 3.2 app that are **not needed**

Lat lon grid

Features that **are needed**

Bulls-eye: grab a real-time point
Mouse icon: grab a pointed at point
Range and Bearing tool
Vehicle heading needs to be displayed
Adjustable bread crumb trail
All properties setting in one window
Be able to use the tool without a data feed
Only 2 vehicles (Ship & ROV)
Option to take alternate data feeds (i.e. from other ships)

Timeline

Prototype for Barry or Paull dive in December
Deployed to WF by GOC: Feb 2008

ROVGISUpgradeMeeting3Oct2007

This page last changed on Oct 03, 2007 by [mccann](#).

Changing ArcMapNav settings

This page last changed on May 18, 2011 by [mccann](#).

Should it be necessary to change a setting that ArcMapNav uses, you can do that by directly editing the user.config file which is located in a directory such as this:

```
C:\Documents and Settings\mccann\Local Settings\Application Data\ESRI  
\DefaultDomain_Path_pz0lsv5r40zaegw4she55pczc5ncbj1k\9.3.1.1850
```

(Replace 'mccann' with the appropriate login, the DefaultDomain_Path_xxxxxx folder may be a different name, and the final subdirectory will depend on the version of ArcMap being used.)

AUV Science Data Processing

This page last changed on Jun 03, 2021 by [kgomes](#).

 **IMPORTANT:** This document has been moved to the following location: <https://docs.mbari.org/internal/se-ie-doc/systems/auvctd/auv-science-data-processing/>. In order to update the document, you need to checkout the source material from this repository <https://bitbucket.org/mbari/se-ie-doc/src/master/>, edit the files and then check them back in. They will be deployed automatically

AUV Science Data Processing

This document describes the Processing steps of Autonomous Underwater Vehicle data within MBARI's Shore Side Data System. It assumes familiarity with general AUV mission execution procedures and the operation of CVS checkout of software, building code using Ant, executing Unix (or Windows) command-line tools and Matlab m-files.

Major sections of this document:

- Step-by-step data processing instructions
- Software checkout and repository code management
- AUV Portal
- Dorado Science Data Processing

Step-by-step data processing instructions

Receive notification of new Dorado mission data, usually an email indicating where the log files are located - most of the time the files are copied to mccann's Tempbox but sometimes the data are provided on a memory stick. The data are first copied to Tempbox where the following steps are executed (review the [AUVCTD Processing Data Flow Diagram](#) to familiarize yourself with the process):

1. Log into elvis as ssdsadmin. Actually, you need to login with your own account, then users by issuing 'sudo -u ssdsadmin -i'
2. Create tar ball of mission directory, e.g. **(be sure to replace the example mission/survey file and directory names with the ones for the new data):**

```
<101 elvis.shore.mbari.org /Tempbox/mccann> cd /mbari/Tempbox/mccann
<103 elvis.shore.mbari.org /Tempbox/mccann> tar cvf 2012.268.07.tar 2012.268.07/
2012.268.07/
2012.268.07/SmartSamplerDataLog.log
2012.268.07/Gulper.log
...
```

```
2012.268.07/ctdDriver2.cfg
2012.268.07/gps.log
2012.268.07/m3dmgx1.log
```

```
<105 elvis.shore.mbari.org /Tempbox/mccann> gzip 2012.268.07.tar
```

3. Copy tarball to /usr/local/auv_ctd and run the auvportal which submits the data to SSDS:

```
<255 elvis.shore.mbari.org /Tempbox/mccann> cp 2012.268.07.tar.gz /usr/local/auv_ctd
<256 elvis.shore.mbari.org /Tempbox/mccann> cd /usr/local/auv_ctd
<257 elvis.shore.mbari.org /local/auv_ctd> ./portal AUVCTD2012268 2012.268.07.tar.gz
Tue, 25 Sep 2012 09:27:43 PDT: processConnection() got message:
AUVCTD2012268;2012.268.07.tar.gz
dstDir.getName() = 2012268
Unpacking /usr/local/auv_ctd/2012.268.07.tar.gz
Copying /usr/local/auv_ctd/2012268/2012.268.07/ctdDriver2.log to /mbari/AUVCTD/
missionlogs/2012/2012268/2012.268.07/ctdDriver2.log
Copying /usr/local/auv_ctd/2012268/2012.268.07/gps.log to /mbari/AUVCTD/
missionlogs/2012/2012268/2012.268.07/gps.log
Copying /usr/local/auv_ctd/2012268/2012.268.07/isislog.log to /mbari/AUVCTD/
missionlogs/2012/2012268/2012.268.07/isislog.log
Copying /usr/local/auv_ctd/2012268/2012.268.07/devices.cfg to /mbari/AUVCTD/
missionlogs/2012/2012268/2012.268.07/devices.cfg
...
```

```

DEBUG [Thread-0] (XmlBuilder.java:538) - appendPrimitiveType called with method name
isValidRole on object of class moos.ssds.metadata.DataProducer
DEBUG [Thread-0] (XmlBuilder.java:482) - appendElement called on class
moos.ssds.metadata.DataProducer using method getClass
DEBUG [Thread-0] (XmlBuilder.java:278) - Converting java.lang.Class(class
moos.ssds.metadata.DataProducer) to XML
Marshallled the AUV mission in /mbari/AUVCTD/missionlogs/2012/2012268/2012.268.07 into XML
Attempting to publish AUVCTD /mbari/AUVCTD/missionlogs/2012/2012268/2012.268.07 to SSDS server
jnp://bob.shore.mbari.org:1099/

```

published XML describing mission AUVCTD /mbari/AUVCTD/missionlogs/2012/2012268/2012.268.07 to SSDS server jnp://bob.shore.mbari.org:1099/

If everything goes well in SSDS:Ingest/Ruminate an email message to auvctd@listserver.mbari.org will arrive shortly.
Tue, 25 Sep 2012 09:34:29 PDT: Done receiving AUV mission data.

Note: With the move of the portal to from jeckel to elvis the mailing to auvctd@listserver.mbari.org has stopped working. This needs to be debugged and fixed.

4. Process the LOPC data, e.g.:

```

<109 elvis.shore.mbari.org /u/ssdsadmin> cd ~/dev/auv_ctd/src/py/auv/
<109 elvis.shore.mbari.org /u/ssdsadmin> /bin/bash
<109 elvis.shore.mbari.org /u/ssdsadmin> export PATH="/usr/local/miniconda2/bin:$PATH"
<109 elvis.shore.mbari.org /u/ssdsadmin> source activate auv_ctd
(auv_ctd) bash-4.2$ processLOPC.pl 2013.224.02
(auv_ctd) bash-4.2$ exit

```

5. Once the mission's instrument netcdf files have been created (check `pismo:/opt/ssds/logs/ssds-updateBot.log` for successful processing) the overall Dorado science data processing script can be edited with the new survey to be processed:

```

<102 elvis.shore.mbari.org /u/ssdsadmin> cd ~/dev/auv_ctd/src/matlab/doradosdp/process/
<103 elvis.shore.mbari.org /doradosdp/process> vi reprocess_surveys.m

```

6. Add a line for the new survey to the end of the `reprocess_surveys.m` file, commenting out the previous survey that was processed:

```

% CANON September 2012 - laps around ESP drifter - last one for this campaign
%%AUVprocess_main('AUVCTD', 2012, ['258.00'], server, ssdsLoadFlag, ssdsLoadsJar,
localFlag, localDataDir, localCalDir, notifyEmail, skipMissionProc, [], 'ctd1')

```

```

% CANON September 2012 - laps around ESP drifter - last one for this campaign
AUVprocess_main('AUVCTD', 2012, ['268.07'], server, ssdsLoadFlag, ssdsLoadsJar,
localFlag, localDataDir, localCalDir, notifyEmail, skipMissionProc, [], 'ctd1')

```

7. Execute the Matlab job with a shell script that properly sets the DISPLAY environment and executes `reprocess_surveys.m` with matlab:

```

<104 elvis.shore.mbari.org /doradosdp/process> RunReprocessJob

```

8. Check for error messages. If everything works an email is sent to the `notifyEmail` setting in `reprocess_surveys.m`, usually auvctd@listserver.mbari.org.

Software checkout and repository code management

The Software for both the AUV Portal and the Science data processing are located in the *auv_ctd* module on the CVS server moonjelly.shore.mbari.org. To check the software to a system use your favorite CVS client, or on Unix use these commands:

```

cd <your_development_directory>
cvs -d :pserver:<login>@moonjelly:/home/cvs login
(enter cvs account password)
cvs -d :pserver:<login>@moonjelly:/home/cvs co -d auv_ctd -P auv_ctd
(about a minute wait for the files to be put in your directory)

```

(Replace `<your_development_directory>` with a subdirectory name of your choice and `<login>` with your moonjelly login name.)

If you have already done this software checkout step and want to update your copy of the software from the repository (which you would want to do if someone else on the team has committed some changes to the repository) then you can get these changes with the cvs update command, .e.g.:`cd <your_development_directory>/auv_ctd`
cvs update (Note that if new directories have been added to the repository you will need to repeat the above checkout command.)

If you have made changes in the software and wish to submit them back to the repository for others to use then use the cvs commit command, e.g.:`cvs commit`
(your default editor will open the files that have been changed for you to enter text describing the changes you made)

AUV Portal

Build and start-up

The AUV Portal is the software that accepts data collected by the AUV. It constructs SSDS-compliant metadata, submits this mission metadata to SSDS and copies the data (.log and .cfg files) to the shore-side network storage location. After checking out the software from CVS, build it with Ant and run it (Note: you will need both Apache Ant and Java JDK 1.4 or higher on your system with the ANT_HOME and JAVA_HOME environment variables set).

```
Build it (ie. Type 'ant')
cd to the build/dist directory
type 'portal' (windows or UNIX):

> cd auv_ctd
> ant (wait for a BUILD SUCCESSFUL message)
> cd build\dist
> portal
> java -cp .\activation.jar;.\auvctd.jar;.\ftp.jar;.\jbossall-client.jar;.\junit.jar;.\log4j.jar;.\siam.jar;.\ssds.jar;.\tar.jar;.\xercesImpl.jar;.\xmlParserAPIs.jar;.\xom.jar;.\moos.operations.portal.AUVPortalService
wait for client...
```

AUV data submission to SSDS

To submit AUV mission data to SSDS, log into the vehicle computer and use the adx command to transfer the data to the AUV Portal that you have started with the above directions. The adx syntax is as follows:

```
usage: adx PortalName DataName login pass survey dir \[dir2 ... \]
PortalName is the hostname of the portal
DataName is the hostname where the auv data resides
login is the login account on DataName that the portal will use
pass is the password to the login on DataName
SSDS survey name (e.g., AUVCTD_2003_191)
dir is the FULL path name of the data directory to copy
```

For example, if auv-2-control is the name of the computer running the Portal, mvc-dmo1 is the name of the AUV's computer, login1 and pass1 are the login name and password for the AUV computer, AUVCTD2004028 is the survey name, and /home/dorado1/auv/altex/onboard/logs/2004.028.05 is the location of the data files on the AUV then the command would be:

```
adx auv-2-control mvc-dmo1 login1 pass1 AUVCTD2004028 /home/dorado1/auv/altex/onboard/
logs/2004.028.05
```

Here is a diagram representing the data flow from AUV through the Portal to SSDS. (Option 2 is what is currently used by the auv_ctd Portal.)



Figure 1. Diagram of data flow between AUV, AUV Portal, and Shore Side Data System.

On the AUV computer (where you ran the adx command the screen will look something like this:

```
bash-2.00$ adx 134.89.13.105 mvc-dmo1 dorado1 dorado1 OCC02003237 /home/dorado1/auv/altex/onboard/
logs/2003.237.04
/home/dorado1/auv/altex/onboard/logs/2003.237.04
; cd /home/dorado1/auv/altex/onboard/logs ...
; tar cvf 2003.237.04_data.tar 2003.237.04 ...

a 2003.237.04/abortMplan.cfg 1 Blocks
a 2003.237.04/bluefinBattlog.log 524 Blocks
a 2003.237.04/crossbow.log 35615 Blocks
a 2003.237.04/ctdDriver.cfg 2 Blocks
a 2003.237.04/ctdDriver.log 11179 Blocks
a 2003.237.04/ctdDriver2.cfg 2 Blocks
a 2003.237.04/ctdDriver2.log 11179 Blocks
a 2003.237.04/devices.cfg 4 Blocks
a 2003.237.04/dvl.cfg 7 Blocks
a 2003.237.04/dvl.log 5323 Blocks
a 2003.237.04/DvlData.bin 33617 Blocks
a 2003.237.04/dynamicControl.log 25404 Blocks
a 2003.237.04/events.log 0 Blocks
a 2003.237.04/gps.log 69 Blocks
a 2003.237.04/hydroscatlog.log 7627 Blocks
a 2003.237.04/isuuslog.log 610 Blocks
a 2003.237.04/lisstlog.log 8626 Blocks
a 2003.237.04/missionfile.cfg.pp 11 Blocks
a 2003.237.04/navigation.log 31120 Blocks
a 2003.237.04/parosci.log 7632 Blocks
a 2003.237.04/syslog 11231 Blocks
a 2003.237.04/tailCone.log 7630 Blocks
a 2003.237.04/vehicle.cfg 7 Blocks
a 2003.237.04/workSite.cfg 4 Blocks
a 2003.237.04/2003.237.04_data.tar.gz 71527 Blocks
; gzip -f 2003.237.04_data.tar ...
; mv 2003.237.04_data.tar.gz 2003.237.04/. ...
; Portal message: 134.89.32.26;dorado1;dorado1;OCC02003237;/home/dorado1/auv/altex/onboard/
logs/2003.237.04/2003.237.04_data
.tar.gz
bash-2.00$
```

On the portal computer the screen will display text something like this:

```
Wait for client...
accepted connection
processConnection() - end-of-stream
Wait for client...
processConnection() got message:
134.89.32.26;dorado1;dorado1;OCC02003237;/home/dorado1/auv/altex/onboard/
logs/2003.237.04/2003.237.04_data.tar.gz
Retrieve mission files from AUV host:
134.89.32.26: /home/dorado1/auv/altex/onboard/logs/2003.237.04/2003.237.04_data.tar.gz
Retrieving /home/dorado1/auv/altex/onboard/logs/2003.237.04/2003.237.04_data.tar.gz
All done
dstDir.getName() = OCC02003237
Unpacking C:\dev\auv_ctd\build\dist\2003.237.04_data.tar.gz
Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\2003.237.04_data.tar.gz
to \\Tornado\AUVCTD\OCC02003237\2003.237.04\2003.237.04_data.tar.gz
Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\abortMplan.cfg to
\\Tornado\AUVCTD\OCC02003237\2003.237.04\abortMplan.cfg \[WARNING\!\! Overwriting
existing file\]
Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\bluefinBattlog.log
to \\Tornado\AUVCTD\OCC02003237\2003.237.04\bluefinBattlog.log \[WARNING\!\! Overwriting
existing file\]
```

Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\crossbow.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\crossbow.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\ctdDriver.cfg to \\Tornado\AUVCTD\OCC02003237\2003.237.04\ctdDriver.cfg
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\ctdDriver.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\ctdDriver.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\ctdDriver2.cfg to \\Tornado\AUVCTD\OCC02003237\2003.237.04\ctdDriver2.cfg \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\ctdDriver2.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\ctdDriver2.log \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\devices.cfg to \\Tornado\AUVCTD\OCC02003237\2003.237.04\devices.cfg
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\dv1.cfg to \\Tornado\AUVCTD\OCC02003237\2003.237.04\dv1.cfg
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\dv1.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\dv1.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\Dv1Data.bin to \\Tornado\AUVCTD\OCC02003237\2003.237.04\Dv1Data.bin
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\dynamicControl.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\dynamicControl.log \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\events.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\events.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\gps.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\gps.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\hydroscatlog.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\hydroscatlog.log \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\isuslog.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\isuslog.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\lisstlog.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\lisstlog.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\missionfile.cfg.pp to \\Tornado\AUVCTD\OCC02003237\2003.237.04\missionfile.cfg.pp \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\navigation.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\navigation.log \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\parosci.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\parosci.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\syslog to \\Tornado\AUVCTD\OCC02003237\2003.237.04\syslog
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\tailCone.log to \\Tornado\AUVCTD\OCC02003237\2003.237.04\tailCone.log
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\vehicle.cfg to \\Tornado\AUVCTD\OCC02003237\2003.237.04\vehicle.cfg
 \[WARNING\!\! Overwriting existing file\
 Copying C:\dev\auv_ctd\build\dist\OCC02003237\2003.237.04\workSite.cfg to \\Tornado\AUVCTD\OCC02003237\2003.237.04\workSite.cfg

[illegible]

```

moos.ssds.model.Deployment.listChildDeployments(java.lang.String)", an
access method that requires an argument, SKIPPING
moos.ssds.model.XmlBuilder: Attempted to call "public java.util.Collection
moos.ssds.model.Deployment.listChildDeployments(java.lang.String)", an
access method that requires an argument, SKIPPING
moos.ssds.model.XmlBuilder: Attempted to call "public java.util.Collection
moos.ssds.model.Deployment.listChildDeployments(java.lang.String)", an
access method that requires an argument, SKIPPING
moos.ssds.model.XmlBuilder: Attempted to call "public java.util.Collection
moos.ssds.model.Deployment.listChildDeployments(java.lang.String)", an
access method that requires an argument, SKIPPING
moos.ssds.model.XmlBuilder: Attempted to call "public java.util.Collection
moos.ssds.model.Deployment.listChildDeployments(java.lang.String)", an
access method that requires an argument, SKIPPING
moos.ssds.model.XmlBuilder: Attempted to call "public java.util.Collection
moos.ssds.model.Deployment.listChildDeployments(java.lang.String)", an
access method that requires an argument, SKIPPING
Marshallled the AUV mission in \\Tornado\AUVCTD\OCC02003237\2003.237.04 into
XML
Attempting to publish \\Tornado\AUVCTD\OCC02003237\2003.237.04 to SSDS.
SUCCESS\\!\\! published \\Tornado\AUVCTD\OCC02003237\2003.237.04 to SSDS

```

Note: Detailed command-line instructions for "portaling" AUVCTD mission data into SSDS are available in the README file in the project or in the operational location on jeckel.shore.mbari.org in /usr/local/auv_ctd/ directory. There is also the portal.log file there which logs the output of the daemon process running on jeckel that receives and processes the data from AUV uploads from the vehicle.

The portal submission process creates self-described netCDF files of all the logged data from each instrument. To produce more useful data products the science data processing must be done; this is described in the next section.

Dorado Science Data Processing

Within the auv_ctd software module is the auv_ctd/src/matlab/process directory. This directory contains the m-files for processing the AUV instrument data such that they may be easily used for scientific investigation.

The products include:

- "level-1" data = calibrated & aligned, in native (full) resolution. This is the basis for the quick-look section plots.
- "level-2" data = level-1 decimated to limiting resolution (ISUS) so that we can serve data as [time, lat, lon, depth, variable 1:N] which is commonly needed.

A "byproduct" of the processing is the generation of processing metadata which is loaded into SSDS so that we can track data products and the software that produce them. Here is a graph snippet showing the level-1 and level-2 data products for an AUV survey

To produce these data products check out the auv_ctd software as described above to a system that has Matlab version 6 or higher. From the auv_ctd/src/matlab/doradosdp/process directory start Matlab and execute the AUVprocess_main script. The documentation for this script is:

```

> matlab \-nodesktop
< M A T L A B >
Copyright 1984-2002 The MathWorks, Inc.
Version 6.5.0.180913a Release 13 Jun 18 2002
To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.
>> help AUVprocess_main
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AUVprocess_main.m: front end script to be modified by the AUV
% science data processing person or in an automated way by SSDS.

```

```
%
% $Id: ScienceDataProcessing.html,v 1.3 2004/05/19 23:17:52 mccann Exp $
%
% Usage:
% AUVprocess_main \[ <campaign>, <year>, <start-year-day.mission>, <end-year-day.mission>, ..
%                  <ssds_load_flag>, <ssdsLoads_jar>, <local_flag>, <local_dir> \]
%
% campaign:      Name of particular type of missions (e.g. 'AUVCTD', 'BIOLUM', 'OCCO')
% year:          Year of the missions
% start-year-day.mission: First mission to use for the section (e.g. '338.15')
% end-year-day.mission:  Last mission to use for the section (e.g. '338.18')
% ssds_load_flag:  If 1 then load processing metadata into SSDS
% ssdsLoads_jar:   Name of SSDS loads jar file, (e.g. ssdsLoads-prey.jar)
% local_flag:      Set to 0 to use SSDS DODS urls for input data, 1 for local files
% local_dir:       If local_flag is 1 then use <local_dir> to read the input data
%
%
%
% Examples:
% To process data from SSDS DODS-accessible datasets, loading SSDS metadata:
%   >> AUVprocess_main('BIOLUM', 2003, '338.15', '338.18', 1, 'ssdsLoads-prey.jar', 0, '')
%
% To process data from local files, not loading SSDS metadata:
%   >> AUVprocess_main('BIOLUM', 2003, '338.15', '338.18', 0, '', 1, '/hosts/tornado/auvctd/')
%
% Year-days must have leading zeros, i.e. they must be 3 digits.
%
%
% Files copied to web accessible locations by this function include:
% - Matlab data file (.mat) of the constructed survey
% - A png image file that is a summary of the data in the .mat file
% - An ODV compliant spreadsheet file (.txt)
% - A COARDS-compliant DODS accessible netcdf file (.nc) that contains the same data of the .mat
  file
% - A log file (diary of this script)
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
>>
```

So to process a set of missions into a survey execute with the start and end mission and the proper parameters for accessing the data and loading the metadata into SSDS. For example this command creates a survey from 3 missions using the DODS accessible netCDF files and loads the data into the SSDS server prey:

```
>> AUVprocess_main('BIOLUM', 2003, '245.02', '245.04', 1, 'ssdsLoads-prey.jar', 0, '')
```

Will produce a bunch of output to the screen, which is captured as a diary log file and saved in SSDS. When it is finished you should see the last few lines like:

```
Submitting metadata to SSDS...
done. ProcessRun id = 1802
Processing time: 289 seconds
```

Important note: If changes are made to any of the matlab m-files they should be checked back into the CVS repository before the AUVprocess_main script is run. This ensures that the ProcessRun Software and Resources will have the correct version number references in SSDS. It's easy to commit the changes. Just run 'cvs commit' from the Unix command line before running AUVprocess_main.

Two level 2 products produced by this processing step are a netCDF file of the aligned and ISUS time step sub-sampled variables and an Ocean Data View spreadsheet file. The ODV Spreadsheet file is not as efficient or as capable as a native ODV Collection (a .var file).

If one wishes to work with AUV survey data in ODV then all the files for a survey in \\tornado\AUVCTD\surveys\<year>\odv should be copied to a local directory so that one has write permission to the files.