Oracle Knowledge : Banner Peak Performance info from Univ of Illinois

This page last changed on Dec 20, 2006 by daniel brint.

Banner performance info from University of Illinois, Dan Brint, Dec 19, 2006

⚠ The information contained here comes courtesy of the Univ of Illinois and while it has not been confirmed personally by ITEc in our environment is believed to be of considerable merrit. Please see the appendix section of the email to get a grasp on how **large** U Illinois is.

Knowledge Article purpose

We received the following information courtesy of Don Erwin of Buff State. It contains quite a bit of useful information from the University of Illinois WRT banner performance tuning. Please add your thoughts on the content at the bottom of the page...

Knowledge Details

Email from Scott Harden sharden@uillinois.edu via boracle

We recently completed our Fall start of term high activity period and it was the best ever since going live in 2003. We experienced ZERO downtime, also a first for Fall start of term. We went live with Banner 7.2/Oracle 10g in March and have been working hard ever since to correct both Banner and Oracle issues. It has paid off and now that it is behind us we thought it would be good to share what we have done to make this happen. We will be upgrading to Banner 7.3 at the end of October so we may find more things to tweak. We hope to present at SUMIT in 2007 but that is a long way off so hopefully this note will assist you in your journey with Banner 7 and Oracle 10g.

To briefly recap where we were and how far we've come:

In 2004, we had slowdowns or outages nearly every day for the month of August. At the time we were on Oracle 9i and facing log file sync waits as well as latch waits. The solution to the log file sync waits was to change the /u04 redo log file system to be an unbuffered file system through Veritas Volume Manager. The latch wait issues were partially solved when we upgraded to Oracle 9.2.0.6 and turned on CURSOR_SHARING = SIMILAR as a band aid for literal SQL.

In 2005, we had a couple of slowdowns and one outage when the database locked up due to an Oracle bug under 9.2.0.6 that caused latch waits to spike. We had a patch ready and implemented it when the outage occurred and we remained stable after that until the Banner 7/Oracle 10g project this past Spring.

In 2006, we had **ZERO** slowdowns and **ZERO** outages and some of the best performance ever for the system as a whole!!

While we tweaked many things since March we think we can safely attribute the success this year to 5 things:

1) **Corrected class search code** so that it would not search across multiple terms if institutions were setup to only search one term. During performance testing of registration prior to the go live of Banner 7 we noticed the class searches were taking roughly 60% of the total CPU. This brought the class search timing down to a negligible number. Defect 1-BQAXW was opened for this and fixed as part of student 7.3. This made a very dramatic impact in registration timings.

2) **New APIs from Sungard HE** that removed the literals and replaced them with bind variables to eliminate extra hard parsing in update DML statements. These were put in the Friday before our Monday start of term and they worked perfectly. They were actually 7.3 APIs that were backward compatible to 7.2. These were Sungard delivered patches p1-ple18_stu70300 and p1-px8eb_gen7030. There will be more to follow in the coming months in regards to defect 1-G7KLZ.

3) **Upgraded from Oracle 10.2.0.1 to 10.2.0.2** in June to overcome some CBO bugs.

4) **Turned off Oracle 10g's automatic memory management (SGA_TARGET = 0)** in early May due the database hanging due to what we considered "thrashing" of the SGA. It was resizing itself thousands of times a minute and was at times growing very large and experiencing high latch waits due to the large SGA size. See Oracle Bug.4472338/4466399 V1020 (36) SYSTEM WIDE HANG ON MMON WAITING FOR 'SGA ALLOCATION FORCING COMPONENT GROWTH':

5) **Turned off Oracle 10g's automatic statistics** gathering process and wrote our own. We relied heavily upon articles and books by Wolfgang Breitling and Jonathan Lewis that indicated that you should not create histograms on tables except when the data is skewed. The automatic stats that come out of the box with 10g incorrectly decide when to use histograms and to what degree to sample the data. In general we opted to gather statistics using 100% sampling and SIZE 1 (no histograms). As we saw issues we added histograms to improve performance after examining the table data to identify data skew/popular values. While this has proven to be very successful we are not yet ready to call if perfect. When we create histograms we use SIZE AUTO now and we are currently debating this because Oracle still is not making some good choices in this area. The suggestion on the table is to build the histograms with SIZE 254.

While those were the major changes that probably brought us the most stability there were several other changes that also have helped us with performance since going to Banner 7 and Oracle 10g. Those include:

- Added an index to FGBTRND on FGBTRND_DEFER_GRANT_IND to correct performance issues with FGRACTG and FRRGRNT. We calculate ICR in deferred mode.
- Added AQ_TM_PROCESSES = 0 to Oracle init file to remove/reduce waits associated with an unneeded qmn process
- We pinned more packages/procedures/triggers in memory (see attached SQL script, you can disregard UILLINOIS objects which are locally developed)
- Adjusted parameters Optimizer_index_caching = 100, Optimizer_index_cost_adj = 0, Optimizer_mode = FIRST_ROWS_10 from original Sungard HE recommended defaults.
- Adjusted parameters _unnest_subquery = true, session_cached_cursors = 75, shared_servers = 0
- Added histograms to SSBSECT, SORWDSP, GLBEXTR, SCBCRSE, STVSUBJ over time as performance problems arose with new statistics gathering process
- Turned off Parallel Query (Parallel_max_servers = 0, Parallel_min_servers = 0) which was causing massive "idle" waits on Finance forms that accessed FGBTRNH and FGBTRND which we had partitioned under Banner 6/Oracle 9i. The idle waits severely affected our ability to use Grid Control to find real performance issues.

- Deleted data from GLBEXTR, reorged the table and indexes online, and added a histogram.
- Added an index to SFRRGFE on SFRRGFE_TERM_CODE, SFRRGFE_TYPE, SFRRGFE_MAJR_CODE, SFRRGFE_PROGRAM, SFRRGFE_TERM_CODE_ADMIT. **Caution:** The data in your registration fee table may be much different than our criteria. You should examine the data and determine which columns are populated at your institution before creating an index on your fee table.
- Dropped index SSBSECT_GSCH_INDEX. The CBO was picking this even though we were not using the SSBSECT_GSCH_NAME column. Our Student tables are VPD'd to segregate our 3 campuses and this index was picked solely for it's VPD code. By removing the index the CBO chose the PK which was the best index for the particular query coming in from Student Web Registration.
- Replace ALL_SYNONYM Oracle 10g view with Oracle 9i version. The 10g version allows it to report on synonyms on synonyms. There are none in our environment and the new 10g view has performance issues. Oracle has indicated that replacing it with the 9i view is an acceptable workaround to the performance issues. This had been a performance issue with our Banner Extender Document Imaging (OTG) process since going live. For some unknown reason it was making thousand of calls per hour to the ALL_SYNONYM view and chewing up 6% of the Oracle database load consistently.
- Eliminated unneeded Oracle VBS processing by removing any VBS policies that were not needed and had no users assigned to them. These entries still remained in the VBS tables, only the actual Oracle policies were removed. They can be re-added at a later date if needed.
- Existing efforts are being made to cleanup and tune Financial Aid RORRULES. There are thousands of these in our environment and this is a large effort to get a better handle on these and have them run as efficiently as possible in our 10g environment.
- Increased the number of jobsub pipes from 1 to 3 in our production environment. This has seemed to eliminate perceived slowdowns for forms users launching jobs in the background waiting in line for access to the jobsub pipe.
- Worked with Sungard HE to fix the gsposec.sql (gubobj policy function) to improve performance. This was made baseline for General 7.3.

Initially we did implement some RULE hints in some forms and packages associated with the FAIINVL, FGIBDSR, and PERJOBS processes. These were added before we redid our statistics gathering. It appears in recent testing that all the RULE hints can safely be removed now that our statistics are consistent. These will be addressed during our 7.3 upgrade.

We have also made HEAVY use of the new Oracle 10g Grid Control tool to monitor performance, drill down on problems, run ADDM, ASH and AWR reports to look for performance issues, and many other great features. It isn't perfect but we've found it to be an invaluable tool in rooting our performance issues.

Lastly, I must give credit where it is due. We have a **SUPER** staff that has worked together to accomplish all of this. The list is too long to name each person but they include our functional and technical analysts, the team that manages the Banner application, DBAs, **and many folks at Sungard HE.**

Submitted on behalf of all those good people by:
Scott Harden
Data Management Area Manager/AITS
University of Illinois

Appendices:

For reference we are **running our database on a SUN F15K with 44 CPUs and 88 GB of memory.** Our database is roughly 300 GB in size. We have around 70,000 students across all 3 campuses and approximately 40,000 employees. We have over **5000 defined Banner forms users and during a typical day will see roughly 1800 to 2000 logged in to the database.** This does not include Web For student or employee users. During Fall start of term we peaked at roughly 1140 concurrent self service users on top of the normal forms load.

Discussion

Please use this area to place comments/thoughts without messing with the original email above.

Dan's thoughts

- They are HUGE 44cpu, 88gb ram, 70,000 students!
- I would hesitate to do things like shutting off the SGA automagic sizing until I saw it first hand. my understanding was that you should NEVER see 1,000's of resizes per minute. I'd want to investigate that myself
- The bit about statistics and histograms looks interesting too

Conclusion

Links to Knowledge Content

Knowledge Links

Procedure HOW to from OTN
Link 1
Link 2

Knowledge Documents

Below are documents collected that are pertinent to this knowledge article.

	Name	Size	Creator	Date	Comment	
	Prod_Parms_20060906.txt	19 kb	Daniel Brint	Dec 20, 2006	Prod init.ora	Edit | Remove
	pin_objects.sql	57 kb	Daniel Brint	Dec 20, 2006	Pinning objects.sql	Edit | Remove
	SETA_Taming_Banner_7.ppt	722 kb	Daniel Brint	Dec 20, 2006	Seta Presentation	Edit | Remove