Speeding Up Traversing of the UniVerse Lock

Abstract

Reducing the page rate of the HPUX when Traversing the UniVerse Lock Table

Solution

On HPUX modifying the size of the TLB (transaction lookaside buffer) for the program that created the disk shared memory segment can reduce the amount of paging on HPUX on an executable by executable basis. On UniVerse, this is done that by altering the parameters on DBsetup and uvsh.

The end effect is the reduction in HPUX page rate "significantly" when searching for record and group locks.

It can be set on an executable by executable basis that way. This could end up with the memory block the disk shared memory segment in just one of the look-aside pointers. This would avoid having 100s of KB in little 4k pages for look up and expiring the relatively small TLB cache would force many memory lookups to go through the slower 4k block cache lookups.

From Benchmarks that have been performed on Large HPUX systems we have seen performance improvements.

For UniVerse it can be implemented in the following way (NB Assumption here is /u2/uv is where UniVerse was installed):

```
/usr/bin/chatr +pd L /u2/uv/bin/DBsetup
/usr/bin/chatr +pd 1M /u2/uv/bin/uvsh
```

And can be verified by using

```
/usr/bin/chatr /u2/uv/bin/DBsetup
/usr/bin/chatr /u2/uv/bin/uvsh
```