DLM (Distributed Large Memory): Remote Memory Paging for Efficient Use of Memory Resource on Clusters

Hiroko Midorikawa, Seikei University, midori@st.seikei.ac.jp, http://www.ci.seikei.ac.jp/midori/paper/paper.html

DLM offers a virtual large memory using distributed node memories in a cluster for multithreaded applications (OpenMP and pthread programs).

Remote paging for semi-Parallel and Pseudo seq. codes to accelerate the performance of seq. codes
- Easy parallelizing by OpenMP
- Usage of implicitly multithreaded library functions

User threads are dynamically created and destroyed.

Simple Threads Suspension : user-level implementation
(1) The DLM hooks a pthread_create() call and replaces it with the tailored pthread_create() call which registers all user thread IDs.
(2) When a user thread accesses the remote data, the com-thread requests and receives a remote page to/from a memory server in background.
(3) Suspends all user thread with pthread_kill() and copy the page in a receive-buffers to user-space.
(4) Restart all user threads using pthread_kill().

AAPC: Automatic Adaptive Page-Size Control for Remote Memory Paging

Fixed-size page transmission sometimes introduces drastic performance degradation.

Estimation of Working Data Set Size in Applications

Criterion for Page Size Control
(1) If WS Page count > Local Page count, a thrashing occurs. Change page size small as closely as the following Target Size.
   Target Size = Local Memory Size / WS Page count
(2) If WS Page count <= Local Page count, there is no thrashing. Change page size little larger to expect more efficient communication, e.g. double the current page size.

Adaptive Page Size Changing
(a) Unified multiple basic pages are used in page size changing
(b) AAPC supports a transmission of a transitional fragmented page generated when page size is changed from small to large.
(c) AAPC allows coexistence of different page size transmission.

Experimental Results of AAPC

Or use dlm library functions for dynamic allocation

Basic Idea of AAPC
Choose an appropriate page size to depress page thrashing between calculation node and memory server nodes. The AAPC repeatedly estimates a working data set size and changes page size dynamically and adaptively to each processing part of an iterative application during it is running.

2 functions for estimating working data set size for each processing part
- void swapin_countstart(int id);
- void swapin_refresh(int id);

Example codes
for( ; ; )
  if( !swapin_countstart( pid ));
  void swapin_refresh( int id );

swapiin_countstart() function
- Calculates target page size using WS page count
- Regulates the target page size of id
- Returns swapin page record.

swapiin_refresh() function
- Estimates WS based on swapin page records
- Calculates target page size using WS page count & regulates the target page size of id
- Returns swapin page record.

Typical Execution Command

matrix_mult 4x4 4 prog args -o matrixes

manufest

- Calculates target page size using WS page count & regulates the target page size of id
- Returns swapin page record.

Inconsistent Page problems in user-level remote memory paging

At low Local Memory Ratio (5%-10%),
- The worst example: NPB BT, SP, FT
- Big page introduces drastic degradation
- Typical example: Himeno Benchmark
- Big page introduces better performance

Local Memory Ratio = Local memory size / Total program data

It is difficult to determine the optimal page size before program running.

AAPC: Automatic Adaptive Page-Size Control for Remote Memory Paging

Fixed-size page transmission sometimes introduces drastic performance degradation.

Estimation of Working Data Set Size in Applications

Criterion for Page Size Control
(1) If WS Page count > Local Page count, a thrashing occurs. Change page size small as closely as the following Target Size.
   Target Size = Local Memory Size / WS Page count
(2) If WS Page count <= Local Page count, there is no thrashing. Change page size little larger to expect more efficient communication, e.g. double the current page size.

Adaptive Page Size Changing
(a) Unified multiple basic pages are used in page size changing
(b) AAPC supports a transmission of a transitional fragmented page generated when page size is changed from small to large.
(c) AAPC allows coexistence of different page size transmission.

Experimental Results of AAPC

Or use dlm library functions for dynamic allocation

Basic Idea of AAPC
Choose an appropriate page size to depress page thrashing between calculation node and memory server nodes. The AAPC repeatedly estimates a working data set size and changes page size dynamically and adaptively to each processing part of an iterative application during it is running.