

Making OpenMP threads abort a for-loop based search - e.g., when a particular thread finds a match such as $SP \% P_n == 0$, just quit and finish up the main program

Hello,

A few of you asked about how to make OpenMP for loop searches abort when a particular thread is successful. Useful for problems like trial division search for an SP factor or for example an associative array search.

Note that you must set an environment variable or OpenMP won't abort because abort support adds overhead, although it appears small for most uses. You can in fact use the environment variable to see the impact. To enable aborts you must add an export to your .bashrc:

OMP_CANCELLATION=true

The code is here - https://www.ecst.csuchico.edu/~sbsiewert/csci551/code/Assoc_Mem (also added to the GitHub <https://github.com/sbsiewertcsu/numeric-parallel-starter-code>)

This is not optimal code (a hash or hash + binary tree would be more optimal), but shows the value of parallel search even for sub-optimal linear search with no hash key

Example output when run on cscigpu with 16 workers:

```
(ultralitics-env) sbsiewert@cscigpu:~/Assoc_Mem$ make
gcc -O3 -g -fopenmp -c assocmem.c
assocmem.c: In function 'find_pattern':
assocmem.c:52:20: warning: 'cancel for' inside 'nowait' for construct
   52 |         #pragma omp cancel for
      |         ^~~~
gcc -O3 -g -fopenmp -o assocmem assocmem.o
(ultralitics-env) sbsiewert@cscigpu:~/Assoc_Mem$ printenv | grep OMP_CANCEL
OMP_CANCELLATION=true
(ultralitics-env) sbsiewert@cscigpu:~/Assoc_Mem$ ls
assocmem assocmem.c assocmem.o Makefile
(ultralitics-env) sbsiewert@cscigpu:~/Assoc_Mem$ ./assocmem 16
16 threads will search for 999999999
Start assoc memory init...
Initialized assoc_mem with sequence up to 1000000000 in 0.173639 secs
Start assoc memory pattern search...
thread 15 found pattern 999999999
thread 14 sees abort
thread 2 sees abort
thread 5 sees abort
```

```

thread 11 sees abort
Search complete with index = 999999999 for size=1000000000
Found pattern=999999999 in assoc_mem at 999999999 in 0.205422 secs
Start assoc memory RAND init...
Initialized assoc_mem with sequence up to 1000000000 in 105.350223 secs
Start assoc memory pattern search...
thread 11 found pattern 561467687
thread 1 sees abort
thread 13 sees abort
thread 7 sees abort
thread 8 sees abort
thread 15 sees abort
thread 3 sees abort
thread 4 sees abort
thread 5 sees abort
thread 9 sees abort
thread 2 sees abort
thread 14 sees abort
thread 10 sees abort
thread 12 sees abort
thread 0 sees abort
Search complete with index = 697729535 for size=1000000000
Found pattern=561467687 in assoc_mem at 697729535 in 0.033920 secs
(ultralitics-env) sbsiewert@cscigpu:~/Assoc_Mem$

```

Note that in both cases, some threads were done with their search before the abort was asserted by the thread that found the pattern requested in the array (note that knowledge of the sequence was not used as can be seen that searching a randomly initialized array is not significantly slower than searching the full sequenced array, which is in fact worst case $O(n)$).

If we test scaling, we see a speed-up of $T_{seq}/T_{parallel} = 3.2543/0.1732 = 18.8x$, for $S=16$, likely $16 < S < 32$ given superscalar cores.

What is also interesting, is that if we have # of workers $< S$, we generally see no aborts, but as we approach # workers $\Rightarrow S$, we start to see aborts, which makes sense since more threads are not done with their search by the time thread "n" finds the pattern at the end of the associative array of 1 billion numbers. Here's some example output:

```

(ultralitics-env) sbsiewert@cscigpu:~/Assoc_Mem$ ./assocmem 1
1 threads will search for 999999999

```

Start assoc memory init...

Initialized assoc_mem with sequence up to 1000000000 in 1.651571 secs

Start assoc memory pattern search...

thread 0 found pattern 999999999

Search complete with index = 999999999 for size=1000000000

Found pattern=999999999 in assoc_mem at 999999999 in 3.254321 secs

(ultralytics-env) sbsiewert@cscigpu:~/Assoc_Mem\$./assocmem 8

8 threads will search for 999999999

Start assoc memory init...

Initialized assoc_mem with sequence up to 1000000000 in 0.255102 secs

Start assoc memory pattern search...

thread 7 found pattern 999999999

Search complete with index = 999999999 for size=1000000000

Found pattern=999999999 in assoc_mem at 999999999 in 0.409962 secs

(ultralytics-env) sbsiewert@cscigpu:~/Assoc_Mem\$./assocmem 16

16 threads will search for 999999999

Start assoc memory init...

Initialized assoc_mem with sequence up to 1000000000 in 0.173184 secs

Start assoc memory pattern search...

thread 15 found pattern 999999999

thread 14 sees abort

thread 1 sees abort

Search complete with index = 999999999 for size=1000000000

Found pattern=999999999 in assoc_mem at 999999999 in 0.205349 secs

(ultralytics-env) sbsiewert@cscigpu:~/Assoc_Mem\$./assocmem 32

32 threads will search for 999999999

Start assoc memory init...

Initialized assoc_mem with sequence up to 1000000000 in 0.184985 secs

Start assoc memory pattern search...

thread 31 found pattern 999999999

thread 28 sees abort

thread 25 sees abort

thread 23 sees abort

thread 30 sees abort

thread 22 sees abort

thread 24 sees abort

thread 19 sees abort

thread 21 sees abort

thread 20 sees abort

Search complete with index = 999999999 for size=1000000000

Found pattern=999999999 in assoc_mem at 999999999 in 0.205522 secs

(ultralytics-env) sbsiewert@cscigpu:~/Assoc_Mem\$