Running automatic optimization with PerfExpert on a matrix-multiplication code

This code is based on the LLNL matrix multiply available at: https://computing.llnl.gov/tutorials/openMP/samples/C/omp_mm.c

To enter the node:

```
$ ./reserve
```

There are four directories with examples. Let's try the 3rd example: parallel matrix multiply using OpenMP.

```
$ cd 3/
```

To try PerfExpert and MACPO in this example using 16 threads, run the following command:

```
$ OMP_NUM_THREADS=16 perfexpert -s mm_omp.s mm_omp
```

PerfExpert will run the automated optimization cycle, generate a new (optimized) source code in the file: `new_mm_omp.c`, and start the second optimization cycle. In the second optimization cycle, PerfExpert will not find a suitable automatic optimization, thus it will show the analysis report (for both performance counters and memory access pattern) and the list of recommendations.

Let's analyze both source codes to check the differences between the optimized and unoptimized versions:

```
$ cat mm_omp.c
$ cat new_mm_omp.c
```

Note the two inner loop indexes on the compute function have been interchanged. Enter the temporary directory to see the report for both codes:

```
$ cd perfexpert-temp-XXXXXX
```

Note the XXXXXX is a random generated sequence which is different on each execution of PerfExpert.

There are two directories, one for each optimization cycle. Let's compare PerfExpert analysis report for each cycle:

```
$ cat */analyzer2.output.txt
```

Note:
- the total runtime shown on each report
- compare the performance metrics on each report

Now, let's analyze the MACPO report for each cycle:

```
$ cat */macpo.output.txt
```

Note:
- the estimated cost to access each variable
- the stride access of each variable
- the reuse factor at different cache levels of each variable

Now leave the node:

```
$ exit
```