

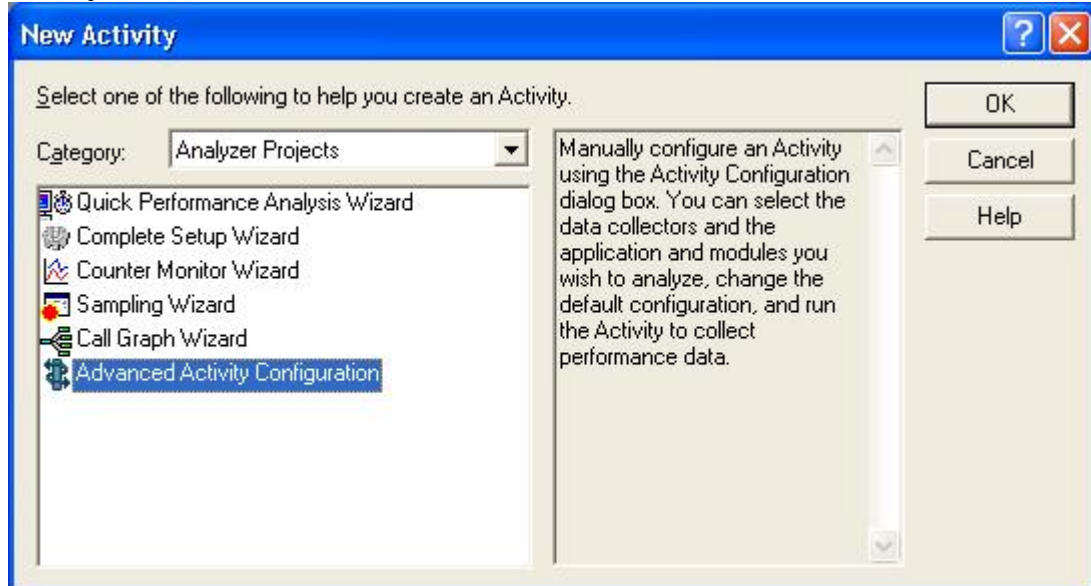
Threading the application

LAB # 3 (Using Thread Profiler)

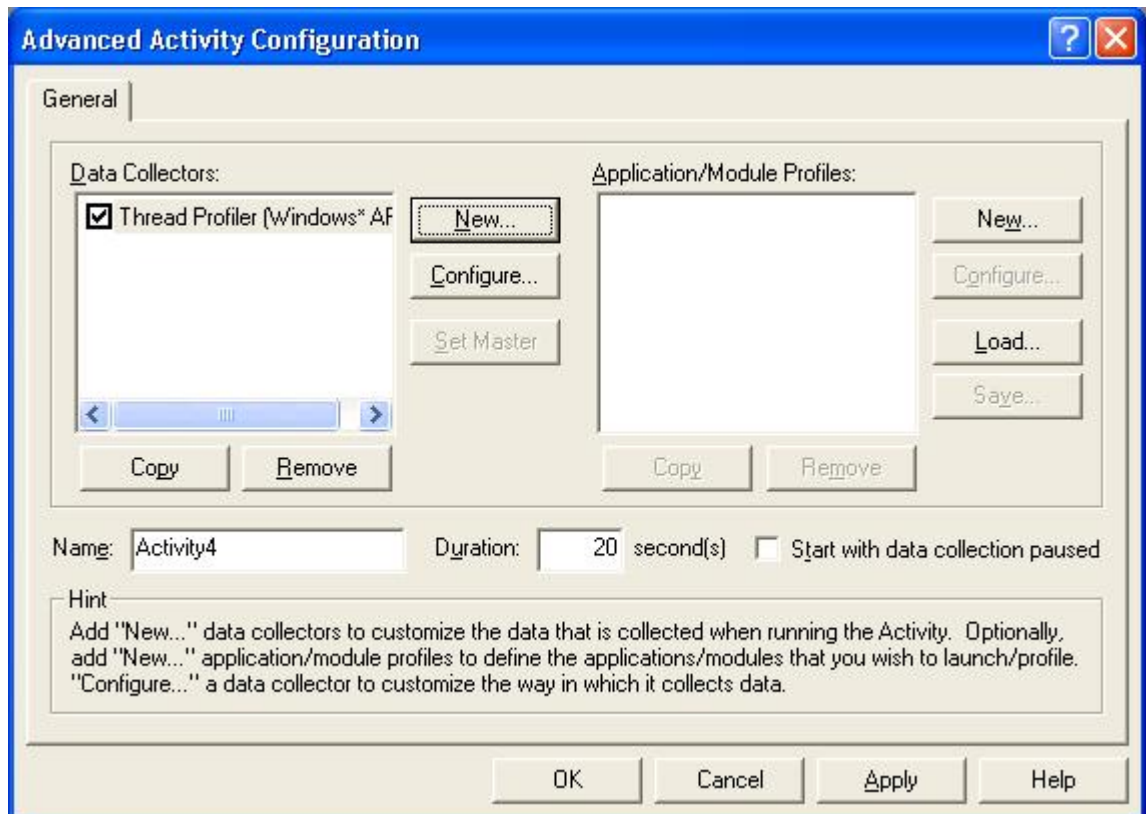
```
////////////////////////////////////
// Copyright (C) 2005 Intel Corp.
//
// Subject to the terms and conditions set forth below, Intel hereby
// grants you a nonexclusive, nontransferable license, to use,
// reproduce and distribute the example code sequences contained
// herein, in object code format, solely as part of your computer
// program(s) and solely in order to allow your computer program(s) to
// implement the multimedia instruction extensions contained in such
// sequences solely with respect to the Intel instruction set
// architecture. No other license, express, implied, statutory, by
// estoppel or otherwise, to any other intellectual property rights is
// granted herein.
//
// ALL INFORMATION, SAMPLES AND OTHER MATERIALS PROVIDED HEREIN
// INCLUDING, WITHOUT LIMITATION, THE EXAMPLE CODE SEQUENCES ARE
// PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY OR
// OTHERWISE, AND INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OF
// MERCHANTABILITY, NONINFRINGEMENT OR FITNESS FOR ANY PARTICULAR
// PURPOSE.
//
// THE MATERIALS PROVIDED HEREIN ARE PROVIDED WITHOUT CHARGE.
// THEREFORE, IN NO EVENT WILL INTEL BE LIABLE FOR ANY DAMAGES OF ANY
// KIND, INCLUDING DIRECT OR INDIRECT DAMAGES, LOSS OF DATA, LOST
// PROFITS, COST OF COVER OR SPECIAL, INCIDENTAL, CONSEQUENTIAL,
// DAMAGES ARISING FROM THE USE OF THE MATERIALS PROVIDED HEREIN,
// INCLUDING WITHOUT LIMITATION THE EXAMPLE CODE SEQUENCES, HOWEVER
// CAUSED AND ON ANY THEORY OF LIABILITY. THIS LIMITATION WILL APPLY
// EVEN IF INTEL OR ANY AUTHORIZED AGENT OF INTEL HAS BEEN ADVISED OF
// THE POSSIBILITY OF SUCH DAMAGE.
//
////////////////////////////////////
```

1. Compiling and running PrimeOpenMP.exe
 - a. The scaling of PrimeOpenMPSolution-1 is low compared to theoretical scaling and is unacceptable. We need to use Thread Profiler to analyze the performance issue.

- b. Open VTune and create a new activity. In “New Activity” dialog choose “advance activity creation”



- c. In “Advance Activity Creation” chose “Thread Profiler (Windows* API)” as data collector.



d. In “Applicatin/Module Profiles” choose “new” and fill the entries as given below.

Application/Module Profile Configuration

General

Application to Launch

Filename: C:\ThreadingClassIDF\PrimeOpenMP\PrimeOpenMPSolution-1.exe

☒ Add to Modules of Interest List

Working directory: C:\ThreadingClassIDF\PrimeOpenMP\

Command line arguments: 1 1000000

System on which to launch application: <localhost>

Local filename for remote application:

☒ Add to Modules of Interest List

Modules of Interest

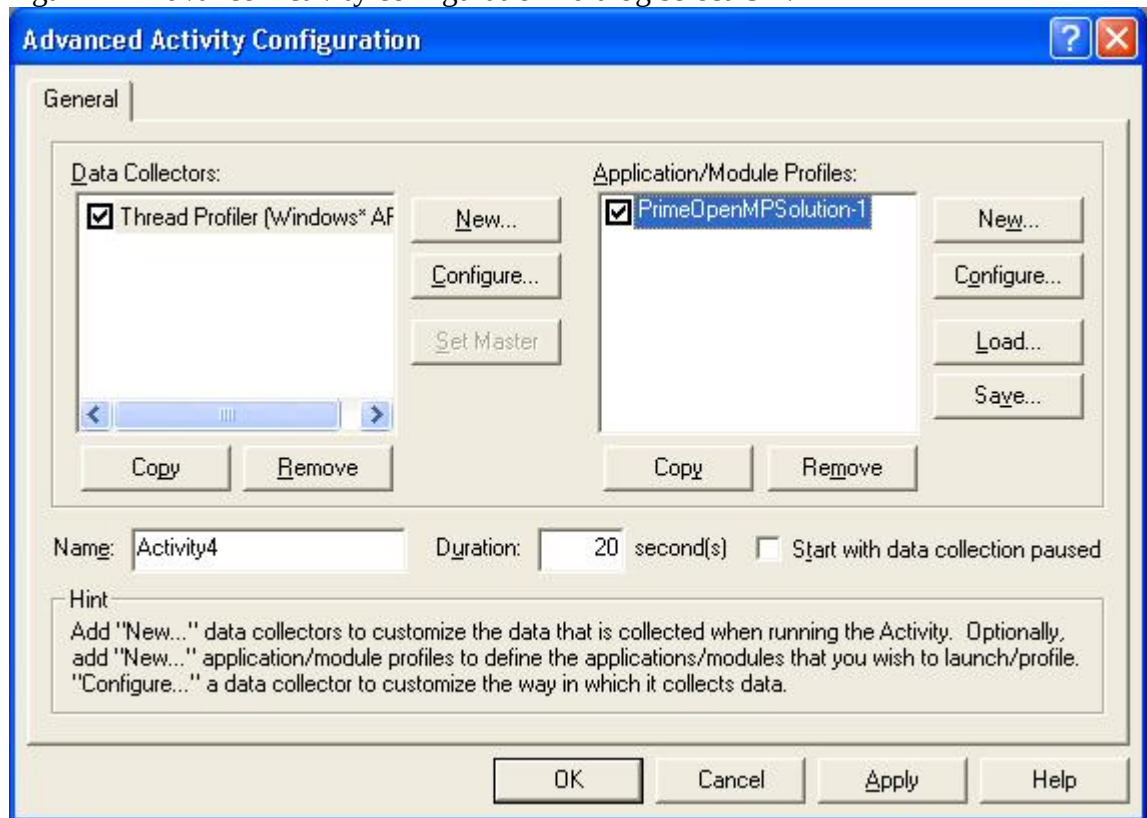
Files: C:\ThreadingClassIDF\PrimeOpenMP\PrimeOpenMPSolution-1.exe

Name: PrimeOpenMPSolution-1

OK Cancel Apply Help

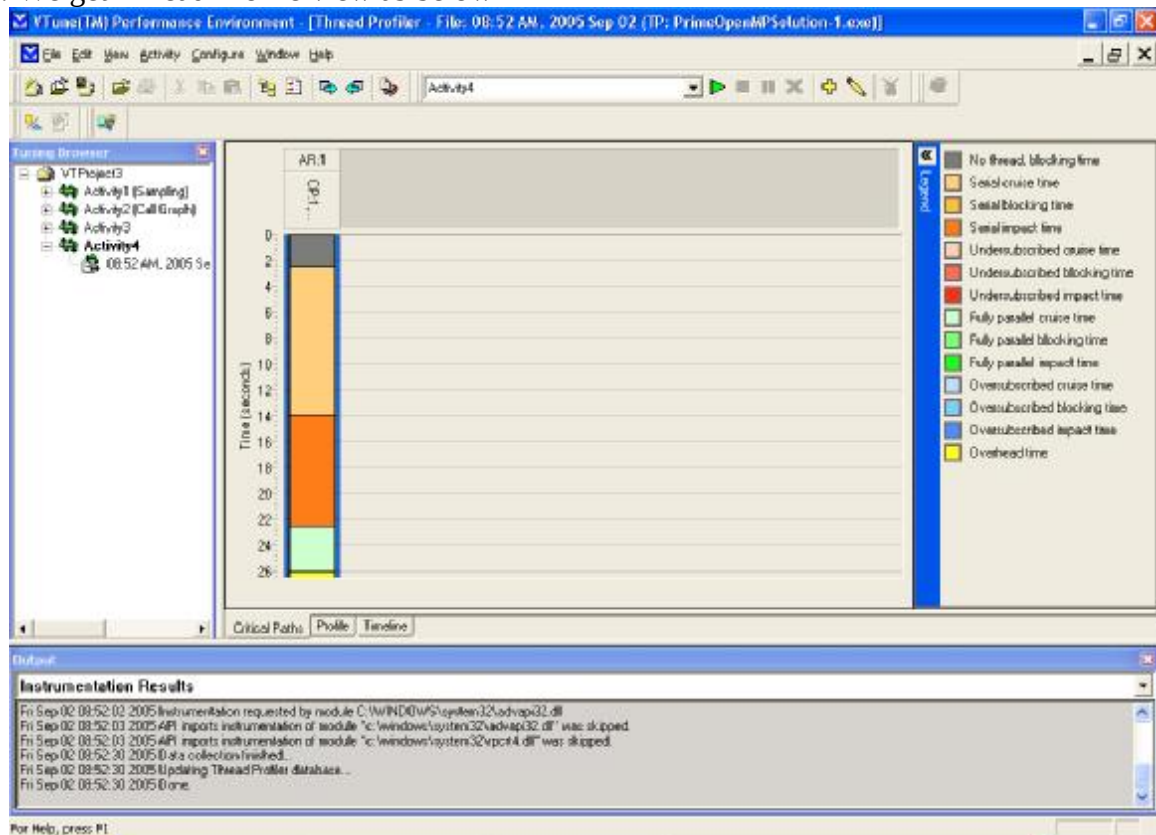
e. Click OK.

f. Again in “Advance Activity configuration” dialog select OK.

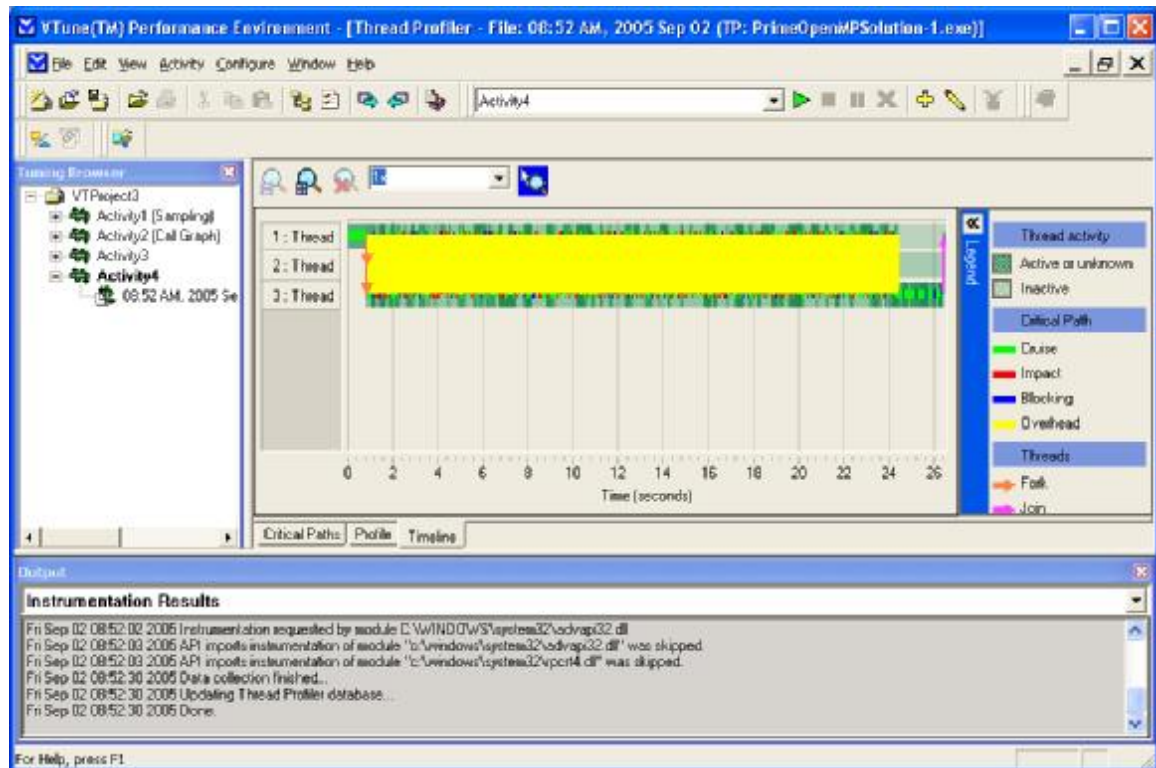


g. Run the activity.

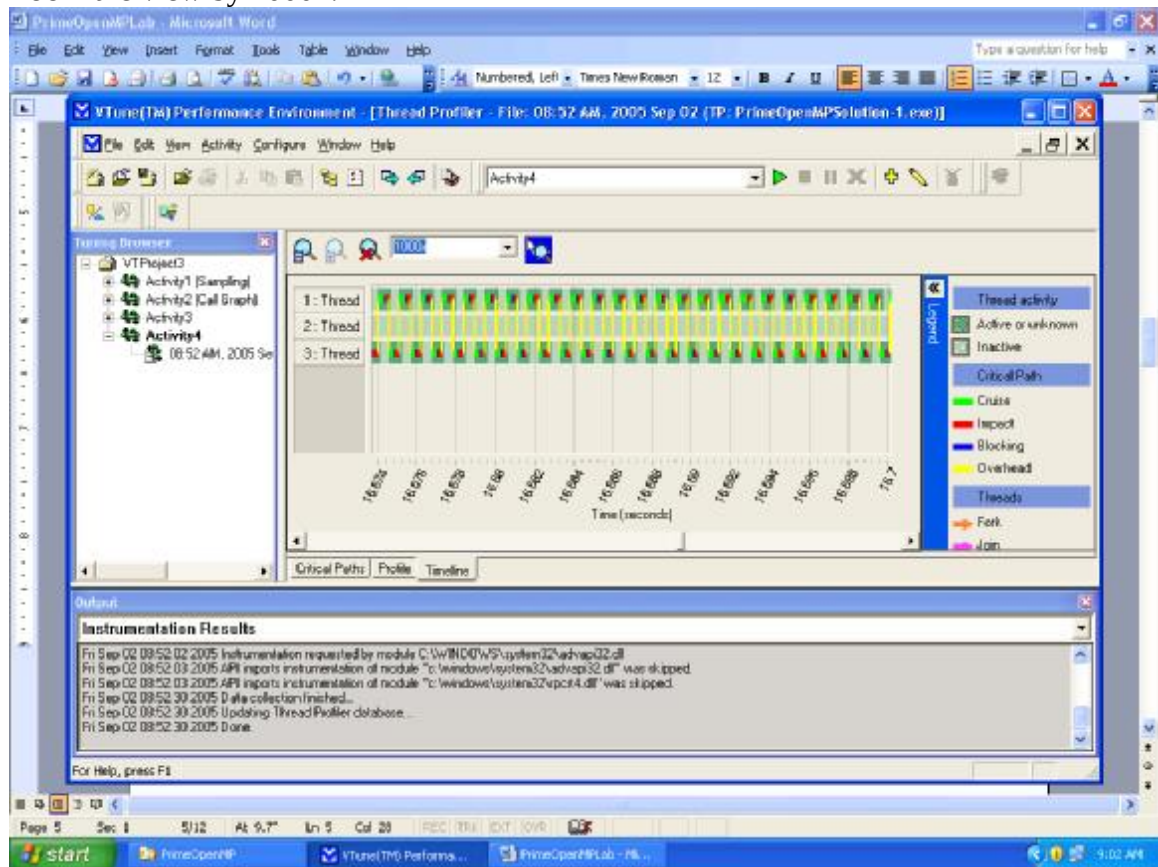
h. We get Thread Profile view as below



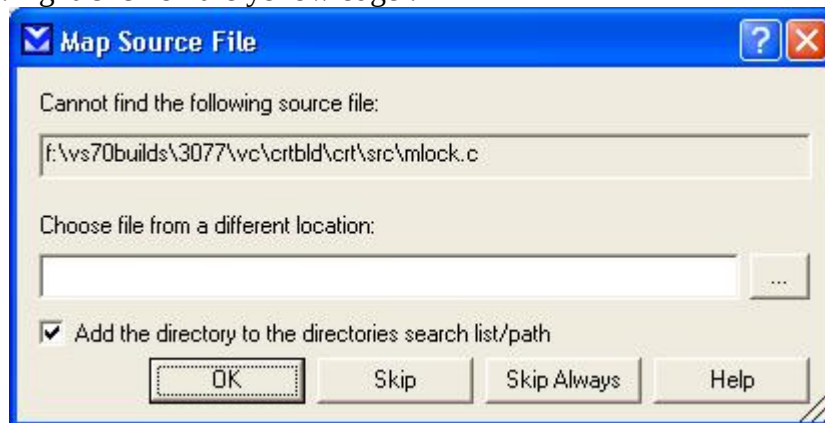
i. Select "timeline" view.



j. Zoom the view by 1000x.



k. Right click on the yellow edge :



l. Click skip

m. You get the call stack. Follow the call stack and locate the first function that is in your application

n. You will see this is “printf()” called from ShowProgress().

o. Fix the ShowProgress() to update only 10 time in application life time of the application.

2. The modified program is given in PrimeOpenMPSolution-2.cpp.
3. Build and run the application the application.
4. Again run the Thread Profiler.
5. View the timeline view

6. Record the number of primes found
7. Record the run time
8. Calculate the scaling factor achieved
9. Compare it with theoretical scaling.