

Find time consuming regions of code in PrimeSingle LAB # 1 (Locating time consuming Regions)

```
////////////////////////////////////  
// 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. Compile and run PrimeSingle.exe
 - a. Invoke the command window for Intel compiler for 32 bi applications from desktop.
 - b. Move to c:\labday[1|2]\ThreadingClassIDF\PrimeSingle directory
 - c. Compile the PrimeSingle.cpp by bld.bat in the Intel C++ compiler command line environment.
 - d. PrimeSingle.exe 1 1000000

```
C:\ Build Environment for IA-32 applications
Volume in drive C has no label.
Volume Serial Number is A0FD-7524

Directory of C:\labday1\ThreadingClassIDF\PrimeSingle
10/04/2005  10:58 AM    <DIR>          .
10/04/2005  10:58 AM    <DIR>          ..
09/01/2005  03:22 PM                89 bld.bat
09/30/2005  05:15 PM                44 clean_bld.bat
09/30/2005  02:52 PM           4,342 PrimeSingle.cpp
               3 File(s)          4,475 bytes
               2 Dir(s)  32,950,083,584 bytes free

C:\labday1\ThreadingClassIDF\PrimeSingle>bld

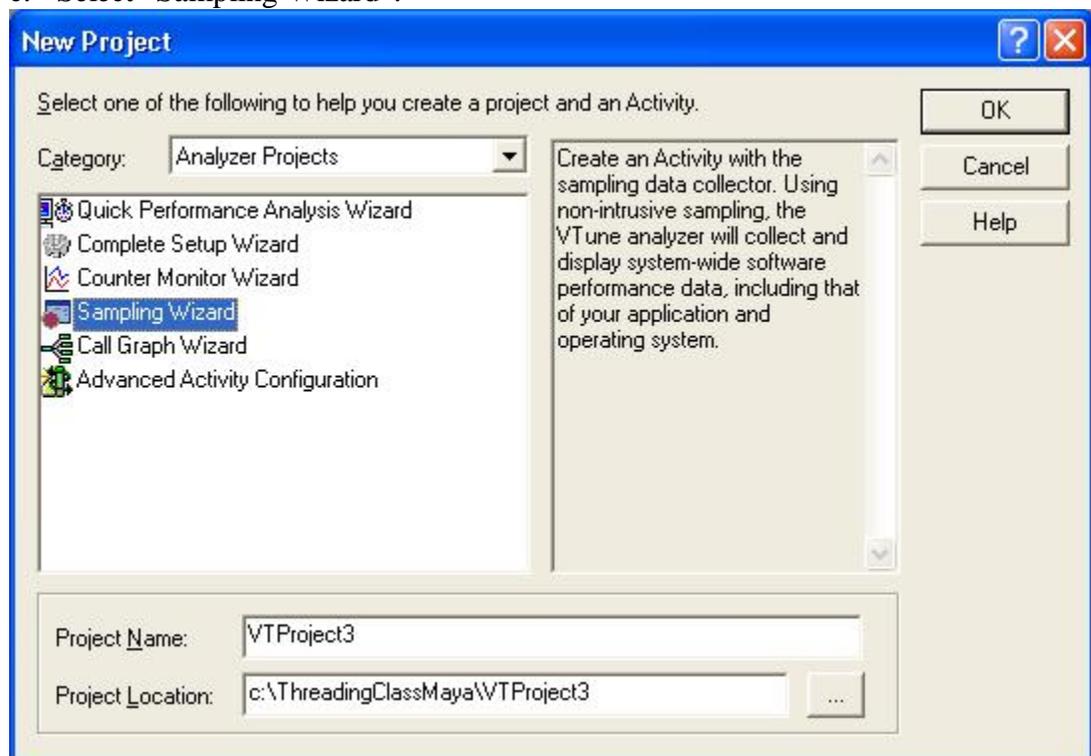
C:\labday1\ThreadingClassIDF\PrimeSingle>icl /O2 /nologo /I ../../include -D WIN
32 /W3 /Zi /Qoption,link,/fixed:no PrimeSingle.cpp
PrimeSingle.cpp

C:\labday1\ThreadingClassIDF\PrimeSingle>PrimeSingle.exe 1 1000000
100%

       78498 primes found between          1 and 1000000 in      3.13 secs

C:\labday1\ThreadingClassIDF\PrimeSingle>
```

- e. Record the number of primes found
- f. Record the run time
2. Analyze the PrimeSingle.exe using VTune to find time consuming regions
 - a. Run Vtune.
 - b. Select “New Project” project.
 - c. Select “Sampling Wizard”.



d. Select Next



- e. Select application to launch PrimeSingle and command line arguments as given below

Sampling Configuration Wizard for Windows*/Linux* Environment (St... ? X)

Machine Name: <localhost> Remote ...

Application

Application to launch: No application to launch
C:\ThreadingClass\DF\PrimeSingle\PrimeSingle.exe ...

Command line arguments:
1 1000000

Working directory:
C:\ThreadingClass\DF\PrimeSingle\ ...

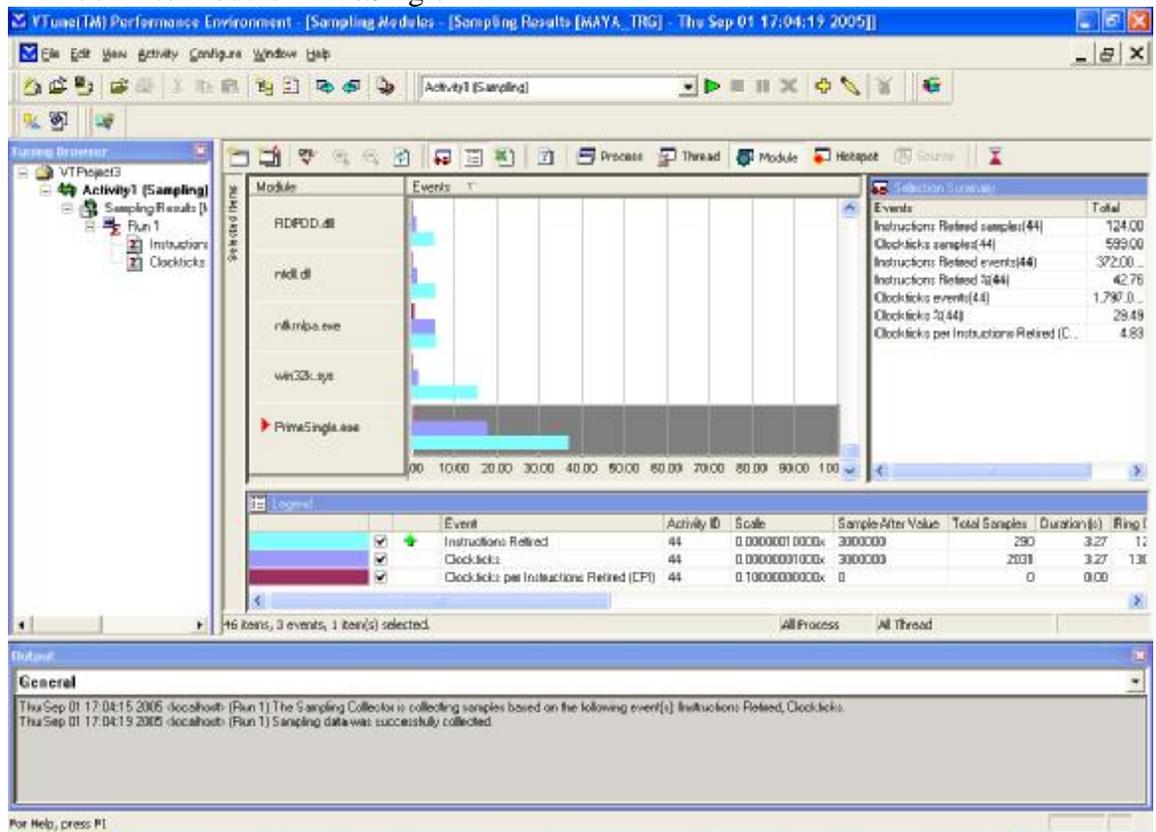
Modify default configuration when done with wizard
 Run Activity when done with wizard

Hint:
Specify the executable that launches the modules you wish to analyze.

< Back Next > Finish Cancel Help

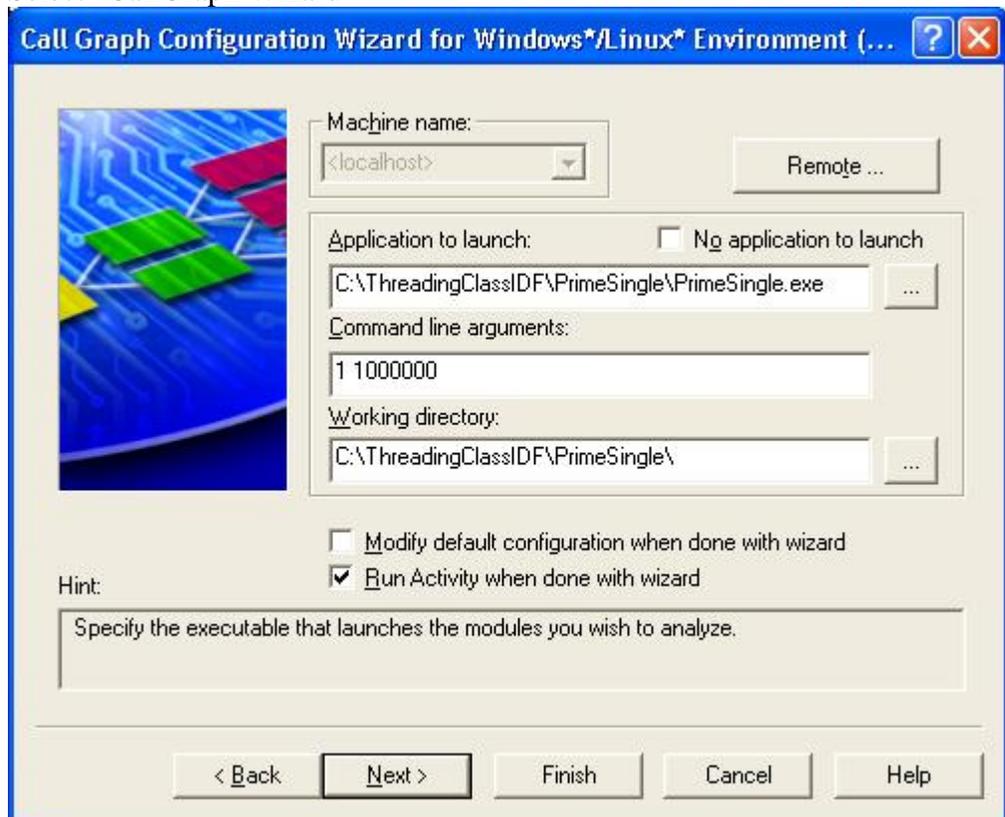
- f. Click finish.

g. Drill down to module PrimeSingle



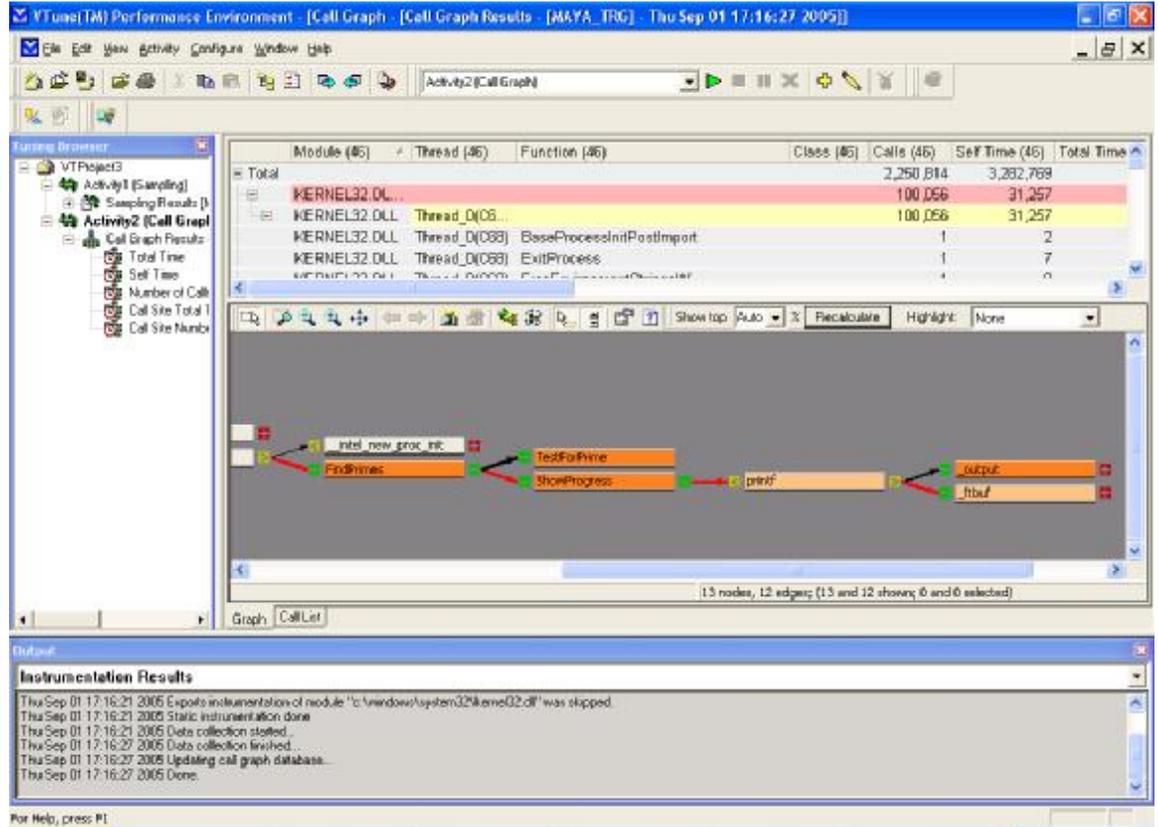
- h. Select only “clockticks” from legend. The Vtune view show TestPrime() is taking highest amount of time when application is run.
3. Find out which function is calling TestPrime()
 - a. Create a new activity in VTune.

- b. Select “CallGraph Wizard”



- c. Click finish.

- d. Scroll to TestPrime() and you will find TestPrime() is called by FindPrime()



4. Calculate the maximum theoretical scaling using two processor and two threads and record