X3Profiler User Manual

Introduction to Profiling

What is profiling?

Profiling allows you to see where in the code your program spends its time. With this information, you can determine which parts of your program are slower than expected or not as efficient as possible. In addition, profiling will also show function calls to other functions during execution. But what information is shown in the profile is affected by how your program is run. If some feature of your program is not used while it is being profiled, some desired feature-specific profile information will not be generated.

Why profiling?

Profiling allows a unique way of analyzing the code of your program. Since the information a profiler collects is gathered during actual execution, it is invaluable for programs that are too large or complex to analyze by just looking at the source code alone.

You can target the areas of code that you need to rewrite to make your program execute faster with the time data. With the function call information, you can see which functions are called more than others, and if they are contrary to your expectations, you can spot bugs that may have gone unnoticed.

Gprof Basics

To compile a source file for profiling, specify the `-pg' option when you run the compiler, using something like cc to do the linking. For example, in your makefile or on the command line,

```
cc -g -c myprog.c utils.c -pg
cc -o myprog myprog.o utils.o -pg
```

Compiling and linking your program with profiling enabled will be enough to generate a "flat profile," which is what X3Profiler needs.

For more information on gprof, please visit:

http://www.gnu.org/software/binutils/manual/gprof-2.9.1/gprof.html

Basic Features of X3Profiler

Note: Our software and its features are based on gprof. That is to say, you must have a gprof file generated before you use X3Profiler.

The Advantage of X3Profiler

When you profile your code using gprof, you will get a raw text output of profiling data. Depending on the length and complexity of your code, the profile may be just as hard to

understand. What X3Profiler aims to do is allow an easier way to look at and understand this profiler by presenting it in a more organized and visually simplistic way. Foremost, the raw profile data file is parsed into a more orderly format internal to X3Profiler called an Atom. The features will allow you to track the efficiency progress of your code along your process, as you are able to save the different graphs and operations you perform on each Atom. The highlight of it all is all the graphing options that allow a more visually intuitive way to view all profiling information, a exponential step above gprof's cluttered text output.

What is an Atom?

In X3 Profiler, an Atom is defined as the simplest form of relevant data to the user. It is generated when a user opens up a raw profile data file that has just been generated by gprof. An Atom will contain information regarding a profiled trial run of a particular program. Once opened, the atom will be the starting point for graph plotting, function operations, and code efficiency history saving.

Creating an Atom

The most important thing is to make sure you have a profile data file already generated. Once that is done, click on the open button or select "Open" from the File menu, to open up the profile. Select the file and you will be prompted to choose a name for your Atom. Now that the Atom is named, you can choose a graph or an operation for the file.

What is a Collection?

Once you have created a series of Atoms, you may want to group them together. Then you can create a named entity for a set of Atoms, called a Collection . This grouping of Atoms aids the user in organizing the data they are working with. It also allows the user to apply a set of operations to the entire collection at once.

Creating/Editing a Collection

Once you have created a series of Atoms that you want to put in a Collection, click on the menu button “New Collection”. You will be prompted to select from a list of Atoms that exist in the directory. Once you have selected the Atoms, you will be asked to name your collection. You can now graph and perform operations on a group of atoms and save all this work under the Collection. If you create more Atoms and what to add them to the Collection, click “Edit” button on the bottom of the left panel, and you will be prompted select the Atom to choose.

Displaying Graph Data

You have a choice of seven graphs to choose from:

1. Bar Graph
2. Line Graph
3. Error Bar Ticks
4. Bar Graph/Compare
5. Stacked Bars/Compare
6. Tick Chart
7. Bar Graph Differentials

You also have choice of five different sets of data to choose from, called Operations:

1. Number of Calls
2. % time
3. Cumulative Seconds
4. Self ms/call
5. Total ms/call

Once you have opened up either an Atom or Collection, click on the “Graph” button in the lower left and you will be prompted to choose one of the seven graph types and one of the Operations. The graph will display on the right panel after you have made your choice, as well as a legend on the bottom, indicating which functions are attributed to which part of the graph.

Menu Reference

File->
1. New
2. Open
3. Save
4. Save As

Top Bar Menu Buttons
1. New Atom
2. New Collection
3. New Operation

Left Panel Tabs
1. Atom
2. Collection
3. Operation

Bottom Left Panel Buttons
Graph -> Graph data from the Atom
Edit -> Change Graph options and Operations or add Atom