

Ctprof

```
{{#switchcategory:MSP430=<McuhitboxHeader/>|C2000=<McuhitboxHeader/>|Stellaris=<McuhitboxHeader/>|TMS570=<McuhitboxHeader/>|MCU=<McuhitboxHeader/>|MAVRK=<MAVRKHitboxHeader/>|<HitboxHeader/>}}
```

Contents

Introduction to ctprof

History

Release 1.0

Installation

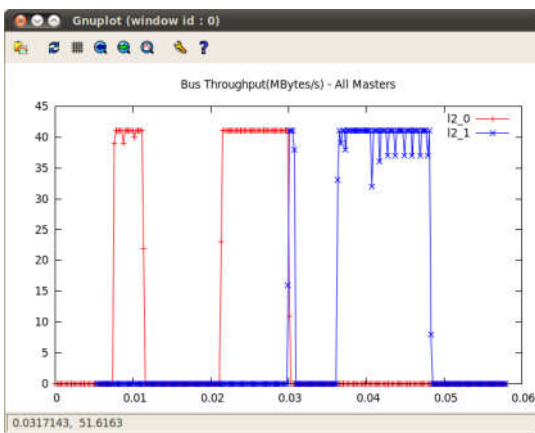
Emupack Installation with CCS
Emupack Installation independent of CCS
gnuplot Installation
Server Installation

References

Examples

TCI6614 DSP L2 Profile Example - using ctprof_sync.sh
TCI6614 DSP L2 Profile Example - using ctprof_srv with pipes

Introduction to ctprof



Bandwidth in MB/S of L2_0 and L2_1 from an ARM example application

For TI devices that support profiling capabilities, the ctprof utility makes using these capabilities significantly easier through a simple use-case based command line utility. Designed with efficiency in mind, the user simply needs to select the connection method, the device, the profile use-case and use-case options on the command line. Ctprof takes care of making the connection, setup of the selected use-case within the target, collection of the trace data, and formatting of the trace data. Data can be formatted for display by CCS (tdf files), as a csv file, or for display with gnuplot (see example gnuplot display left).

Designed as a host command line utility, ctprof supports different methods to communicate with a target depending on the environment. If the target is running Linux, a server (ctprof_srv) that runs on the target device's ARM core can provide Ethernet communications to the host's client (ctprof) utility. For bare-metal applications an XDS may be used to provide communications with the target.

See the History section below for communication methods supported, target devices supported, and target environment requirements per release.

History

Release 1.0

The first release of ctprof provides the following support:

- Supports devices TMS320TCI6614 and TMS320TCI6612
- The ctprof utility is only available for a Linux host with this release
- The Linux server application (ctprof_srv) utilizes TCP/IP sockets to communicate with the host utility (ctprof). A pre-built version of ctprof is delivered with the SC-MCSDK 2.2.0 release (pending) or the source can be cloned from a git repository(see the Installation section)
- Requires Emupack 5.1.275.0
- System level bus profiling use cases for TCI6614 and TCI6612 slave modules with a CP Tracer module
- No XDS support

Installation

ctprof is delivered with an Emupack. See one of the following two Emupack sections for your specific situation (Emupack installation with CCS or independent of CCS).

If gnuplot is to be used with ctprof, it must be installed separately. ctprof generates a gnuplot command file that is based on the gnuplot's 4.2 patchlevel 6 release, so it should be compatible with gnuplot versions newer than 4.6. If you use the "-f gnuplot -o output_file" option with ctprof, ctprof will generate a output_file.gnuplot_cmd and output_file.gnuplot_csv files that can be viewed on any system with "gnuplot output_file.gnuplot_cmd".

ctprof_srv and ctprof_ex (example application) are delivered pre-built with the SC_MCS SDK 2.2.0 release or it can be built from source. For building from source see the server installation section.


Emupack Installation with CCS

If you have CCS installed then simply use the built-in "Help->Install New Software" mechanism from within CCS. If the version of the Emupack you need is not provided by this method then you can download the Emupack from [Emupack Packages](#). Please follow the instructions at [Emupack CCS Installation](#)

Emupack Installation independent of CCS


If you are installing the emupack independent of CCS you can download an Emupack from [Emupack Packages](#). Please follow the installation instructions at [Emupack Independent Installation](#).

For installation on a Linux machine that does not already have CCS installed, see [Linux Host Support](#) for notes on supported versions of Linux hosts and versions we would expect to work, but do not officially support.



Useful Tip

If you install the Emupack with root privileges (not required) it will automatically install the ctprof man page at /usr/share/man/man1. If you do not have root privileges then the man page will not be installed, but the install_ctprofman.sh script will be included in ccs_base\emulation\analysis\bin\ . This script can be used by root to install the man page.



Useful Tip

The Emupack installation will not update your PATH environment variable. You must add the following directory to your path manually or cd to this location and execute "./ctprof":

/install_dir/ccsv5/ccs_base/emulation/analysis/bin

gnuplot Installation

For Ubuntu "sudo apt-get install gnuplot" will work. The minimum gnuplot revision required is 4.2 patchlevel 6. For the later versions of gnuplot or for building from source see the [gnuplot download site](http://www.gnuplot.info/download.html) (<http://www.gnuplot.info/download.html>).

Server Installation

If you are not using the pre-built roots from the SC_MCS SDK 2.2.0 release, ctprof_srv can be built from source. These following instructions assume you have the CROSS_COMPILE environment variable set to the compiler that is compatible with the target's Linux run-time environment. It also requires a TCP connection to the target. From your Linux host:

1. git clone git@git.ti.com:ctprof_srv/ctprof_srv.git (This will clone the ctprof_srv repository).
2. cd ctprof_srv/server
3. make all arm DESTDIR=~/.tmp (see the makefile for other build options)
4. scp ~/.tmp/ctprof_srv root@x.x.x.x:/usr/bin

The ctprof_srv project contains ctprof_sync.sh, which is a script that will sync ctprof_srv's state with the execution of your application. From your Linux host:

1. cd ctprof_srv
2. chmod 755 ctprof_sync.sh
3. scp ctprof_sync.sh root@x.x.x.x:/usr/bin

The ctprof_srv project also contains example_app. The example_app program provides a simple set of ctprof_srv synchronization functions that allow for precise data collection. Assuming the CROSS_COMPILE environment variable is set to an appropriate compiler, from your host Linux machine:

1. cd ctprof_srv/example_app
2. make all arm DESTDIR=~/.tmp (see the makefile for other build options)
3. scp ~/.tmp/ctprof_ex root@x.x.x.x:/usr/bin

References

See the [ctprof User's Guide](#) for details on commands and options.

For general information on the CP Tracers and use cases supported by ctprof and CCS see [CP Tracer Details](#).

Examples

Note: these examples assume you have Linux running on your target and you know its IP address.

TCI6614 DSP L2 Profile Example - using ctprof_sync.sh

The ctprof_ex will perform a simple zero memory test over the first 64K of each DSP's L2 space. To profile the accesses from the ARM to the DSP's L2 memory, simply use the ctprof_sync.sh script with the following steps:

1. From the target console start ctprof_srv and ctprof_ex using ctprof_sync.sh script:

```
root@tc16614-evm:~# ctprof_sync.sh ctprof_ex
ctprof_srv:Waiting for client to connect
ctprof_srv:Client command port connected
ctprof_srv:Client etb service port connected
```

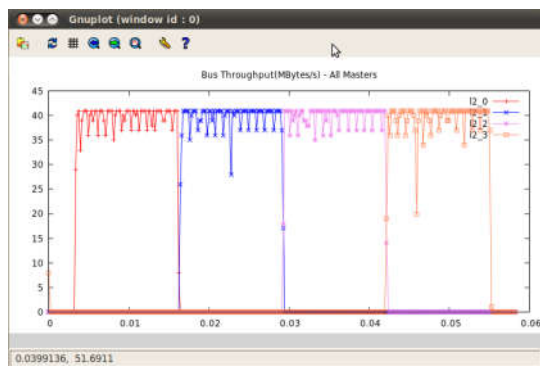
2. From the client execute the ctprof command:

```
ctprof --connect server:x.x.x.x --device tc16614 --duration 60 --sample-rate .2 --format gnuplot bw l2_0 l2_1 l2_2 l2_3 -o l2
```

3. On the target console:

```
ctprof_srv:Opening pipe for writing. Waiting on pipe to be open for reading.
ctprof_srv:pipe open for reading
execute:./ctprof_ex
ctprof_srv:Recording started
ctprof_ex:Test Iterations left 1
ctprof_ex:Testing 65536 words of Corepac L2_0
ctprof_ex:Corepac L2_0 had 0 failures
ctprof_ex:Testing 65536 words of Corepac L2_1
ctprof_ex:Corepac L2_1 had 0 failures
ctprof_ex:Testing 65536 words of Corepac L2_2
ctprof_ex:Corepac L2_2 had 0 failures
ctprof_ex:Testing 65536 words of Corepac L2_3
ctprof_ex:Corepac L2_3 had 0 failures
ctprof_ex:Exiting
ctprof_srv:Recording stopped
ctprof_sync exiting
```

4. On the host the following gnuplot window will be opened:



Bandwidth in MB/S of ARM writes to DSP L2's

5. Hitting "return" in the client console will terminate gnuplot

**Helpful
Tips**

Useful Tip

You can also filter on an address range and on access direction (read or writes). See [Ctprof Users Guide](#) for details.

TCI6614 DSP L2 Profile Example - using ctprof_srv with pipes

This option provide precise control over when ctprof_srv collects profile data by providing synchronization of the server state with the application. This option requires integration of the ctprof_utility.c/.h files with your application. See the ctprof_ex.c source for an example. The [ctprof_srv User's Guide](#) also has details on ctprof_srv pipe operation.

To try this option with ctprof_ex use the following steps:

1. On the target start ctprof_srv in pipe mode:

```
root@tc16614-evm:~# ctprof_srv -P &
ctprof_srv:Waiting for client to connect
```

2. On the target start the ctprof_ex in pipe mode:

```
root@tc16614-evm:~# ctprof_ex -P &
```

```
./ctprof --connect server:x.x.x.x --device tci6614 --op-mode signal --sample-rate .2 --format gnuplot bw l2_0 l2_1 l2_2 l2_3 -o l2
```

```

root@tc16614-evm:~# ctprof_srv:Client command port connected
ctprof_srv:Client etb service port connected
ctprof_srv:Opening pipe for writing. Waiting on pipe to be open for reading.
ctprof_srv:pipe open for reading
ctprof_ex:Test Iterations left 1
ctprof_ex:Testing 65536 words of Corepac L2_0
ctprof_srv:Recording started
ctprof_ex:Corepac L2_0 had 0 failures
ctprof_ex:Testing 65536 words of Corepac L2_1
ctprof_ex:Corepac L2_1 had 0 failures
ctprof_srv:Recording stopped
ctprof_ex:Testing 65536 words of Corepac L2_2
ctprof_ex:Corepac L2_2 had 0 failures
ctprof_ex:Testing 65536 words of Corepac L2_3
ctprof_ex:Corepac L2_3 had 0 failures
ctprof_ex:Exiting

```

Bus Throughput(MBytes/s) - All Masters

l2_0
l2_1
l2_2
l2_3


0.0205179, 51.6163

Bandwidth in MB/S of ARM writes to DSP L2's

```
ctprof_srv:Waiting for client to connect
```

<p>{{</p> <p>1. switchcategory:MultiCore=</p> <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum <p>Please post only comments related to the article Ctprof here.</p>	<p>■ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum</p> <p>■ For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum</p> <p>Please post only comments related to the article Ctprof here.</p>	<p>C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article Ctprof here.</p>	<p>DaVinci=For technical support on DaVinciplease post your questions on The DaVinci Forum. Please post only comments about the article Ctprof here.</p>	<p>MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article Ctprof here.</p>	<p>OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article Ctprof here.</p>	<p>OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article Ctprof here.</p>	<p>MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article Ctprof here.</p> <p>For technical s... please post yo... questions at http://e2e.ti.com Please post on... comments abo... article Ctprof h... }}</p>
---	--	--	---	---	--	---	---

<https://processors.wiki.ti.com/index.php/Ctprof>

	Amplifiers & Linear	DLP & MEMS	Processors	Switches & Multiplexers
	Audio	High-Reliability	<ul style="list-style-type: none">▪ ARM Processors▪ Digital Signal Processors (DSP)▪ Microcontrollers (MCU)▪ OMAP Applications Processors	Temperature Sensors & Control ICs
	Broadband RF/IF & Digital Radio	Interface		Wireless Connectivity
	Clocks & Timers	Logic		
	Data Converters	Power Management		

{{#switchcategory:MSP430=<McuHitboxFooter/>|C2000=<McuHitboxFooter/>|Stellaris=<McuHitboxFooter/>|TMS570=<McuHitboxFooter/>|MCU=<McuHitboxFooter/>|MAVRK=<MAVRKHitboxFooter/>|<HitboxFooter/>}}

Retrieved from "https://processors.wiki.ti.com/index.php?title=Ctprof&oldid=162622"

This page was last edited on 4 October 2013, at 17:03.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.