Streamline使用说明

发布版本:1.0

作者邮箱: <u>cmc@rock-chips.com</u>

日期:2017.12

文件密级:公开资料

前言

概述

产品版本

芯片名称	内核版本
全系列	4.4

读者对象

本文档 (本指南)主要适用于以下工程师:

技术支持工程师

软件开发工程师

修订记录

日期	版本	作者	修改说明
2017-12-25	V1.0	陈谋春	

Streamline使用说明

- 1. 介绍
- 2. 抓取数据

3. 分析结果

1. 介绍

Streamline是DS5提供的一个性能分析工具,通过抓取cpu和gpu的内部硬件计数器和Kernel的一些软件 tracepoint来实现性能分析,功能比较强大。

2. 抓取数据

- 准备工具
 - 1. 首先需要下载ds5,一些新的cpu和gpu可能需要较新的DS5才能支持,我目前用的是5.26版本。DS5是 需要license的,不过可以先申请一个30天的全功能试用license,下载安装和申请license的流程就不在 这里介绍了。
 - 2. 设备端需要和host建立连接,目前支持两种连接方式:adb和网络
 - 3. 需要设备端的root权限,不然没法在设备端运行gatord
 - 4. 需要和固件匹配的符号表, 方便后续的分析
- gatord

设备端需要运行一个守护进程来和host端做交互,早期版本的mali驱动或ds5版本还需要重新编译Kernel的gpu驱动来启用gatord,不过目前新版本的ds5已经没有这个问题了。

ο

1 \$ adb push /path/to/ds5/sw/streamline/bin/\$ARCH/gatord /data/local/ 2 # cd /data/local/ 3 # ./gatord &

• 建立连接

gatord跑起来之后就可以在host端打开Streamline,在左上角点击

Connection Brows	er		
Connection Browser			
Select a target connectio	n		
Select a target connectio			
Streamline Agent via adb	_		
EGP6CYN74U	adb device	gator v5.61	
		Setup Target Cancel Select	

选中你要调试的设备,点"select"即可。

• 配置计数器

Streamline支持的计数器和设备相关,在连接完成后,可以点击

按钮,会弹出对话框,列出所有支持的计数器	竖:		
😣 🗉 Counter Configuration			
Choose the target counters to collect.			
Connected to adb:EGP6CYN74U.			
		×	
Available Events		Events to Coll	ect
Cortex-A53	2 of 6 available 🚊	Cortex-A53	6
Branch: Immediate	=	Branch: Mispredicted	
Branch: Indirect	Ŭ	Bus: Access	
Branch: Mispredicted		Cache: L2 data access	
Branch: PC change		Clock: Cycles	
Branch: Potential prediction		Instruction: Executed	
Branch: Taken		Cortex-A72	
Bus: Access		(4(
Bus: Cycle			
Bus: Read		Event Based Sam	pling
Bus: Write		Threshold	
Cashar Allacata mada			
	Load Defaults	Save	Cancel

左边是可选的计数器,右边是已选的计数器,在可选计数器上双击即可移动到已选计数器,完成后点"save"按钮即可,全部重选可以点击"load defaults"先恢复默认设置。

• 抓取数据

计数器选好以后,可以点击) 开启抓取,结束时点击	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	,这时候会自动跳转到分析界面。
--------------	--------------	---------------------------------------	-----------------

3. 分析结果

Streamline相对于传统的profile工具的优势在于:丰富的硬件计数器支持,可以很方便的看到cache,bus和gpu内部状态。

• 加载符号表

右键单击左侧的我们抓取到的数据名称,在弹出的菜单中点击"analyze",会弹出如下对话框:



Analyze

Choose the settings to produce a new report.

Analysis Ø Process Extra Debug Information (when available)
Resolution mode: Normal ‡
Program Images
 \${streamline_results}/fish_02.apc/images/vmlinux \${streamline_results}/fish_02.apc/images/libwebviewchromium.so \${streamline_results}/fish_02.apc/images/libc.so
Cancel Analyze

点击红色箭头指向的按钮就可以添加带符号信息的elf文件了。不用全部添加,根据热点添加即可。

• Heat Map

这个视图可以很方便的找到热点线程,点击左下角的



点击▶可以展开各个进程,点击每个线程的名字,可以只显示这个线程的统计值,例如线程的cpu占有率,线程的miss rate等。

时间轴上有个滑块,可以拉伸和移动,以显示某个时间段内的统计数据,类似下图:

Stream	line (D	S-5 5.2	6.2)					ð 🖮 🗤	◀)) 18:02 🔱
			😣 🗇 🗉 Streamline						
Q	0°C		🗟 Streamline Data 📃 🗖	😰 Help 🛛 🔂 fish 🔂 fish	02 🖾				ned int) (/
			• 🗱 🌣 adb:EGP6CYN74U 💿 🕐	🔄 Timeline 💋 Call Paths 🔕	Functions 🗟 Code < Log				
			🖑 🗄 🕞 Filter 🛛 🗶 🕞 🖄 🕍	<u>}}}</u>	5ms 🔻	(7.079s	0 🖹 🔻 🖬	🔽 🚳 🖬 🕐	0)
			▶ caffe_gpu	0.85s 0.9s 0.95s	1s 1.05s 1.1s 1.15s 1.2s	1.25s (1.29s [45ms] 35s	1.4s 1.45s 1.5s 1.55s 1.6s	1.65s	- /
			► caffe_gpu_02						a
			▶ caffe_gpu_03	► CPU ACTIVITY (A72)	5 0%	• 49	9.13%		k/arm/boot-
	0°C 08:0	00	► caffe_gpu_04	F System				1	lib/libart.
			► caffe_gpu_05	CPU ACTIVITY (A53)		0.0	00%		
			► caffe_gpu_06	F System				 = (trtuat addr
		0	► fish	 Branch (A72) F Mispredicted 	BOK		18,782		ls from /de
			▼ fish 02					V	
			7s 79ms @ 2017-11-13 16:39:19 DS-5 v5.61	 Branch (A53) Mispredicted 	50 K	00			vik-jit-cod
A		СР	Rockchip RK3399 Evaluation Boar 1ms Resolution		N				
			/home/cmc/Documents/Streamline V	 Bus (A72) Bus (A72) 	200 K				
a			► fish_03			••••••••••••••••••••••••••••••••••••••		~~ ~ ~ ~	
			► tensorflow	► Bus (A53) 🗱	= 50 K				
		内存		Cache (A72)	= 300 K			AAAAAA	
							(0.013) · · · · · · · · · · · · · · · · · · ·	1	
\geq				Cache (A53)	90 K				
6									
				=					
		网络		[idle]					
				 [acr.browser.barebones 					
-~-				{Chrome_InProcRe #14					a the second second second
				{Chrome_InProcGp #15 {PenderThread #1470}					
<u></u>				{mali-cmar-backe #1443}					
14-	•			Jacr browser barebone		Appotation Filter			
				Row Fill	A.1	Annotation Filter		O ACCIVICY V	
		1							

• Core Map

这个视图可以看到每个线程各个时刻都在那个core上跑,对于看调度问题比较方便,比如不合理的cpu迁移。

• Cluster Map

这个视图可以看到每个线程在当前跑在哪个cluster,可以分析是否有不合理的大小核迁移导致性能下降。

• Samples

这个视图可以分析每个时间片内函数的cpu占比,时间片可以通过上面时间轴上的滑块控制,对于分析热点函数比较有用。

• Processes

这个视图可以分析热点进程的cpu占比。

• Functions

点击上部的"functions"选项卡,可以看到整个抓取周期内的函数热点统计,如下图所示:

		Ps	😣 🗇 🗉 Streamline										
0°C			🗟 Streamline Data 👘 🗖	🕼 Help 🛛 🗟 fish	Ish_02 ∞							- 0	jned int)
			● 🖽 🏶 adb:EGP6CYN74U 💿 ⑦	🔜 Timeline 💋 Call P	aths 💊 Functions 🗟 Code ؇ Log								
1			🔗 🕀 🛱 Filter 🛛 🗶 🍋 🖄 🖄	A 3 🔩 🔐 Row Filte	3 🐜 🔂 Row Filter					1 1	1 1,547 (3.62%) 🔬 🧿 🚽		
			h coffe anu		Function Name	Self 🔻	% Self	Total	% Total	Instances	Stack	Size	so)
			r carre_gpu	cpuidle enter state		17,746	41.50%	17.746	41.50%	1	640	788	
			▶ caffe_gpu_02	<unknown code="" in="" lit<="" td=""><td>GLES mali.so></td><td>2,601</td><td>6.08%</td><td>2.601</td><td>6.08%</td><td>8</td><td>0</td><td>2</td><td></td></unknown>	GLES mali.so>	2,601	6.08%	2.601	6.08%	8	0	2	
			► caffe gpu 03	accum_dump_buffer		2,531	5.92%	2,531	5.92%	1	0	80	rk/arm/b
				_raw_spin_unlock_in	3	2,054	4.80%	2,054	4.80%	66	128	64	lib/lib
0.0	08:00		▶ caffe_gpu_04	sk_fill_path(const Sk	Path&, const SkIRect&, SkBlitter*, int, int,	1,547	3.62%	1,547	3.62%	1	3,072	928	,
			► caffe gpu 05	_raw_spin_unlock_ir	grestore	919	2.15%	919	2.15%	52	128	64	an 195 - 2000
				SkAlphaRuns::Break	short*, unsigned char*, int, int)	825	1.93%	825	1.93%	1	80	128	/irtual -
			► caffe_gpu_06	SuperBlitter::blitH(in	t, int, int)	771	1.80%	771	1.80%	1	192	364	ls from
		0	► fish	<unknown code="" ga<="" in="" td=""><td>itord></td><td>527</td><td>1.23%</td><td>527</td><td>1.23%</td><td>5</td><td>0</td><td>2</td><td>JUS TION</td></unknown>	itord>	527	1.23%	527	1.23%	5	0	2	JUS TION
1				<unknown code="" in="" sy<="" td=""><td>stem_server></td><td>435</td><td>1.02%</td><td>435</td><td>1.02%</td><td>13</td><td>0</td><td>2</td><td></td></unknown>	stem_server>	435	1.02%	435	1.02%	13	0	2	
			75 79ms @ 2017-11-13 16:39:19	memcpy		377	0.88%	377	0.88%	7	0	592	so)
				memset		344	0.80%	344	0.80%	8	0	272	.vik-jit
		CP	Rockchip RK3399 Evaluation Boar 1ms Resolution	do_softirq		251	0.59%	251	0.59%	32	1,408	864	
			/home/cmc/Documents/Streamline 🔍	<unknown code="" in="" lit<="" td=""><td>art.so></td><td>231</td><td>0.54%</td><td>231</td><td>0.54%</td><td>2</td><td>0</td><td>2</td><td></td></unknown>	art.so>	231	0.54%	231	0.54%	2	0	2	
7			► fish_03	SkTIntroSort <skedg< td=""><td>e*, SkTPointerCompareLT<skedge>>(void,</skedge></td><td>227</td><td>0.53%</td><td>227</td><td>0.53%</td><td>1</td><td>192</td><td>656</td><td></td></skedg<>	e*, SkTPointerCompareLT <skedge>>(void,</skedge>	227	0.53%	227	0.53%	1	192	656	
		6	· 131_05	SkPathRef::Editor::E	ditor(sk_sp <skpathref>*, int, int)</skpathref>	194	0.45%	194	0.45%	1	128	260	
			 tensorflow 	ipt_do_table		187	0.44%	187	0.44%	5	2,944	1,552	
7				arch_cpu_idle		186	0.44%	186	0.44%	1	384	388	
				SkA8_Coverage_Blitt	er::blitAntiH(int, int, const unsigned char	157	0.37%	157	0.37%	1	96	68	
		内存		dma_clean_range		154	0.36%	154	0.36%	6	0	48	
				divsi3		131	0.31%	131	0.31%	2	0	220	
				pthread_mutex_lock		129	0.30%	131 (0.31%) %	11	0	64	
2		6		.plt [libwebviewchro	mium.so]	119	0.28%	119	0.28%	8	0	4,148	
				pi_memcpy	Long to the second	116	0.27%	116	0.27%	5	0	384	
				<unknown code="" in="" lit<="" td=""><td>hwui.so></td><td>108</td><td>0.25%</td><td>108</td><td>0.25%</td><td>3</td><td>0</td><td>2</td><td></td></unknown>	hwui.so>	108	0.25%	108	0.25%	3	0	2	
				<unknown code="" in="" ke<="" td=""><td>rnel></td><td>100</td><td>0.23%</td><td>100</td><td>0.23%</td><td>1</td><td>0</td><td>2</td><td></td></unknown>	rnel>	100	0.23%	100	0.23%	1	0	2	
				<unknown code="" in="" lit<="" td=""><td>outils.so></td><td>97</td><td>0.23%</td><td>97</td><td>0.23%</td><td>10</td><td>0</td><td>2</td><td></td></unknown>	outils.so>	97	0.23%	97	0.23%	10	0	2	
		PAR		aaa_waik_edges(SKA	nalyticedge", SkAnalyticedge", SkPath::Fl	96	0.22%	96	0.22%	1	928	5,964	
				_raw_spin_lock		95	0.22%	95	0.22%	10	0	08	
				pthread_mutex_unic	ICK	94	0.22%	94	0.22%	9	0	198	
				intee	hinderset	91	0.21%	91	0.21%	9	0	1,088	
				sunknown code in lit	vsomposor rk20board.co>	90	0.21%	90	0.21%	/	0	2	
				<unknown code="" in="" ny<="" td=""><td>vcomposer.rksoboard.s0></td><td>87</td><td>0.20%</td><td>8/</td><td>0.10%</td><td>2</td><td>1 0 2 0</td><td>200</td><td></td></unknown>	vcomposer.rksoboard.s0>	87	0.20%	8/	0.10%	2	1 0 2 0	200	
-1				refresh_cpu_vm_stat	2121013	83	0.19%	83	0.19%	-	1,920	288	
- I) F)	

• Call Paths

如果想看函数的调用关系,可以切换到"call paths"选项卡