Golang 程序性能分析工具安装及使用说明

1. **概述**

Golang程序的CPU、内存消耗分析可以使用工具pprof、go-torch来完成。pprof是用来采集程序运行时CPU、内存消耗样本，保存到本地文件，并根据样本数据生成调用图。go-torch可以根据样本数据生成火焰图。它底层依赖pprof和FlameGraph脚本。本说明主要记录相关工具的安装及基本使用（以CPU消耗为例）。以下环境测试通过：

软件环境：Ubuntu 16.04.2 LTS + go version go1.8.3 linux/amd64

1. **安装**

* pprof：

pprof已被集成到golang开发环境。已经安装golang开发环境之后，可以通过如下命令检测是否已经成功安装：

go tool pprof --help

* go-torch：

下载安装go-torch:

go get github.com/uber/go-torch

go-torch可执行文件会被安装到$GOPATH/bin目录下。可通过如下命令检测是否已经成功安装：

go-torch --help

* FlameGraph：

下载FlameGraph脚本：

git clone https://github.com/brendangregg/FlameGraph.git

为了让go-torch能够访问到FlameGraph脚本，需要将FlameGraph的目录添加到PATH环境变量中。

1. **使用**

* 代码添加pprof支持

如果程序本身对外提供Http服务，则只需要在main package引入pprof包对其进行初始化。pprof会自动在默认Http服务中注册handler，提供的服务都在/debug/pprof路径下。

import(

\_"net/http/pprof"

)

如果程序本身不对外提供Http服务，则需要在程序启动时添加一个Http监听。代码如下：

import(

"log"

\_"net/http/pprof"

)

func main(){

go func() {

log.Println(http.ListenAndServe("localhost:6060", nil))

}()

………

}

完成以上添加，Golang程序本身已经支持了pprof远程样本采集。远程采集方法见下方。

* 采集CPU消耗样本

运行Golang程序。通过如下命令远程连接分析目标进程，进行CPU消耗样本采集：

输入：go tool pprof -seconds 30 http://localhost:6060/debug/pprof/profile

输出：Saved profile in /home/user/pprof/pprof.pb.gz

-seconds 30：指定采集30秒的样本数据。默认30秒，可以根据实际设置。

localhost:6060：应填写目标分析进程实际的Http服务监听地址。

记下输出保存的pb.gz文件路径，后面的图表生成要用到。

* 生成调用图和火焰图

根据前面的pb.gz文件生成调用图，并保存文件wanchain-cpu-call.svg：

go tool pprof -svg /home/lzh/pprof/pprof.pb.gz > "./wanchain-cpu-call.svg"

根据前面pb.gz文件生成火焰图，并保存文件wanchain-cpu-blume.svg :

go-torch -b /home/lzh/pprof/pprof.pb.gz -f "./wanchain-cpu-blume.svg"

* 直接生成分析图表

也可以跳过保存profile(pb.gz)文件，直接生成图表。

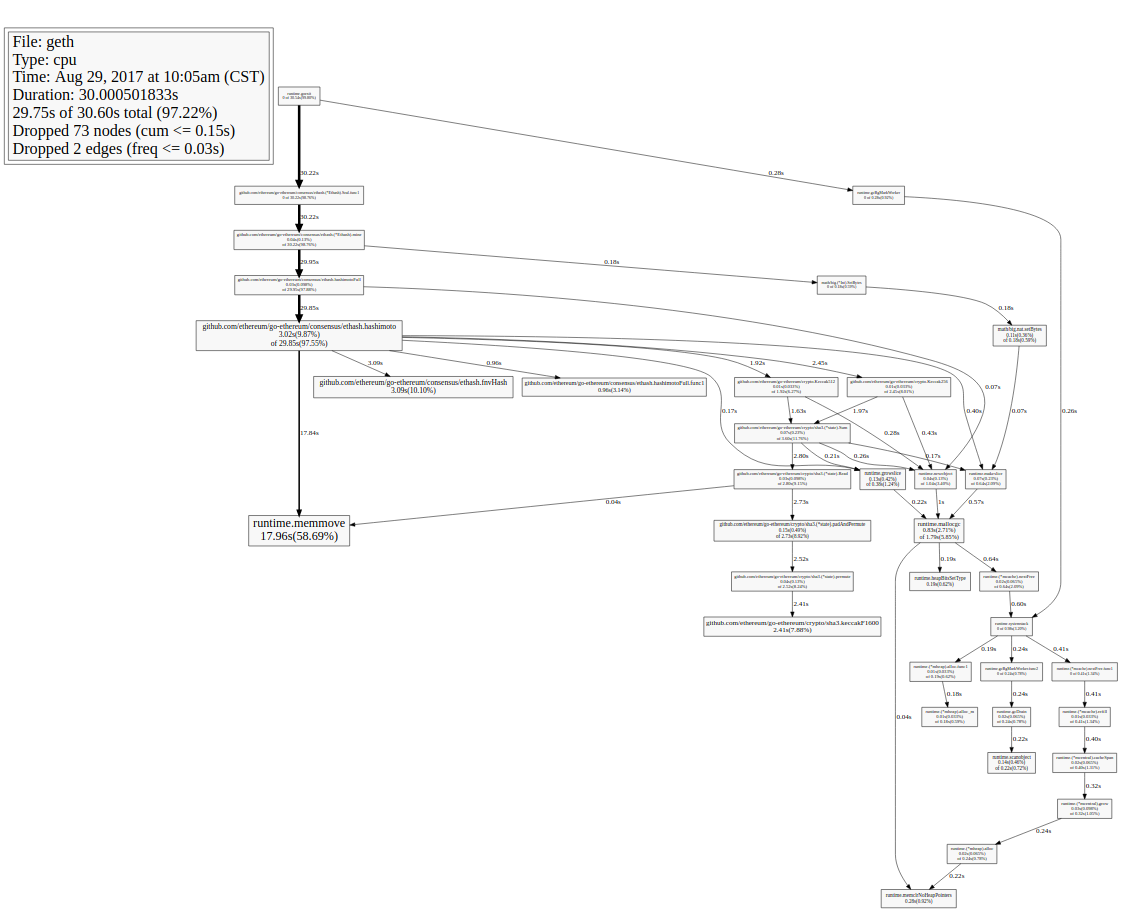
直接生成调用图：

go tool pprof -seconds 30 -svg http://localhost:6060/debug/pprof/profile > "./wanchain-cpu-call.svg"

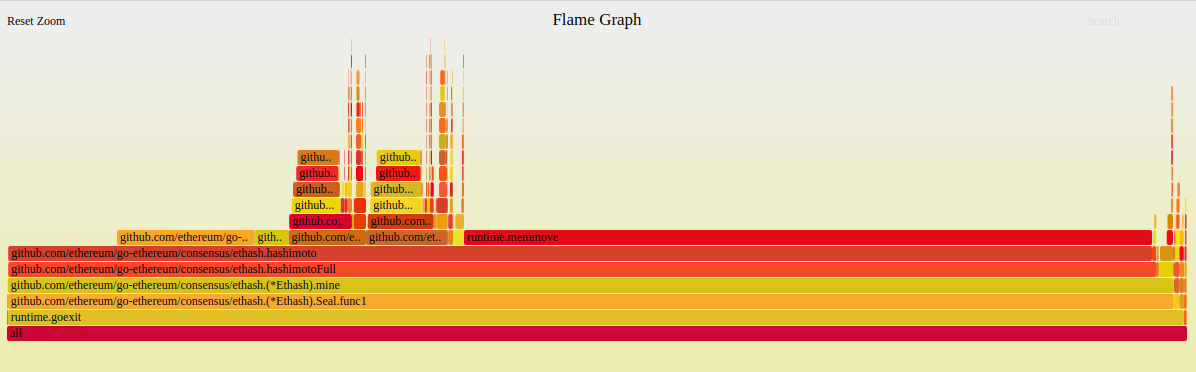
直接生成火焰图：

go-torch --seconds 30 -u http://localhost:6060/debug/pprof/profile -f "./wanchain-cpu-blume.svg"

以上生成的svg文件可以在浏览器中打开查看，效果如下：



（调用图）

（火焰图）

（完毕）