一、Linux内存检测

1. free df dmeg vmstat Massif

vmstat :

进程

内存

分页

阻塞 IO

中断

磁盘

CPU

2、kasan

支持arm64和x86

kasan是一个动态检查内存错误的工具，可以检查内存越界访问、使用已释放内存、重复释放以及栈溢出。

1. valgrind 检测内存泄漏
2. Top怎么查看每个cpu的使用量
3. Process被绑定在哪个cpu上
4. Htop

1.2 工具

Valgrind的最新版是3.11.0，它一般包含下列工具：

1.Memcheck

最常用的工具，用来检测程序中出现的内存问题，所有对内存的读写都会被检测到，一切对malloc()/free()/new/delete的调用都会被捕获。所以，它能检测以下问题：

对未初始化内存的使用；

读/写释放后的内存块；

读/写超出malloc分配的内存块；

读/写不适当的栈中内存块；

内存泄漏，指向一块内存的指针永远丢失；

不正确的malloc/free或new/delete匹配；

memcpy()相关函数中的dst和src指针重叠。

2.Callgrind

和gprof类似的分析工具，但它对程序的运行观察更是入微，能给我们提供更多的信息。和gprof不同，它不需要在编译源代码时附加特殊选项，但加上调试选项是推荐的。Callgrind收集程序运行时的一些数据，建立函数调用关系图，还可以有选择地进行cache模拟。在运行结束时，它会把分析数据写入一个文件。callgrind\_annotate可以把这个文件的内容转化成可读的形式。

3.Cachegrind

Cache分析器，它模拟CPU中的一级缓存I1，Dl和二级缓存，能够精确地指出程序中cache的丢失和命中。如果需要，它还能够为我们提供cache丢失次数，内存引用次数，以及每行代码，每个函数，每个模块，整个程序产生的指令数。这对优化程序有很大的帮助。

4.Helgrind

它主要用来检查多线程程序中出现的竞争问题。Helgrind寻找内存中被多个线程访问，而又没有一贯加锁的区域，这些区域往往是线程之间失去同步的地方，而且会导致难以发掘的错误。Helgrind实现了名为“Eraser”的竞争检测算法，并做了进一步改进，减少了报告错误的次数。不过，Helgrind仍然处于实验阶段。

5.Massif

堆栈分析器，它能测量程序在堆栈中使用了多少内存，告诉我们堆块，堆管理块和栈的大小。Massif能帮助我们减少内存的使用，在带有虚拟内存的现代系统中，它还能够加速我们程序的运行，减少程序停留在交换区中的几率。

此外，lackey和nulgrind也会提供。Lackey是小型工具，很少用到；Nulgrind只是为开发者展示如何创建一个工具。

<https://jingyan.baidu.com/article/948f59247be1c8980ef5f96b.html>

Valgrind 调用上述工具的说明

valgrind在ARM64平台移植说明(以ubuntu为例)

<https://blog.csdn.net/m0_37874806/article/details/105590314>

1、安装说明

(1)、先安装automake sudo apt-get install automake

如果没有将交叉编译工具链添加到环境变量，可以不使用下面的export,

1. #直接在confgure使用CC指定交叉编译工具链的位置也可以

(2)、export CC=aarch64-linux-gnu-gcc

(3)、export LD=aarch64-linux-gnu-ld

(4)、export AR=aarch64-linux-gnu-ar

1. ./autogen.sh
2. ./configure --prefix=/your/path/to/install/valgrind --host=aarch64-linux --enable-only64bi

二、Linux网络检测工具

1、网络性能测试工具，Iperf 灌包工具

测试TCP吞吐量

iperf是以C/S模式工作，运行于服务端和客户端下。服务端主要用于监听到达的测试请求，而客户端主要用于发起连接会话。因此要使用iperf，需要两台服务器，一台运行在服务器模式下，一台运行在客户端模式下

是一个基于Client/Server的网络性能测试工具，可以测试TCP、UDP和SCTP带宽质量，能够提供网络吞吐率信息，以及震动、丢包率，最大段和最大传输单元大小等统计信息，帮助我们测试网络性能，定位网络瓶颈。其中抖动和丢包率适应于UDP测试，带宽测试适应于TCP和UDP。

1. 怎么用

iperf详解

<https://www.cnblogs.com/klb561/p/9215952.html>

|  |  |
| --- | --- |
| 命令行参数 | 含义描述 |
| -s | 将iperf以server模式启动，例如：iperf3 –s，iperf3默认启动的监听端口为5201，可以通过“-p”选项修改默认监听端口 |
| -D | 将iperf作为后台守护进程运行，例如：iperf3 -s - |

### **2.3.3 Iperf应用实例**

要使用iperf，首先要启用一个服务端，这里假定服务端的IP地址为192.168.12.168，在此服务器上运行“iperf3 -s”即可开启iperf的服务器模式。在默认情况下，iperf3将在服务端打开一个5201监听端口，此时就可以将另一台服务器作为客户端执行iperf功能测试了。

(1)

（1）TCP方面

q 测试网络带宽。

q 支持多线程，在客户端与服务端支持多重连接。

q 报告MSS/MTU值的大小。

q 支持TCP窗口值自定义并可通过套接字缓冲。

（2）UDP方面

q 可以设置指定带宽的UDP数据流

q 可以测试网络抖动值、丢包数

q 支持多播测试

q 支持多线程，在客户端与服务端支持多重连接。

2、抓包工具 tcpdump

3、磁盘检测工具Iotop

4、网络监视工具 iftop

2、

* ****tcpdump**** : 用于抓取tcp/ip包, 分析网络问题的必需神器
* ****netstat****: 查看网络状态
* ****netcfg****: 配置网口(使能网口以及配置IP等)
* ****iptables****: 网络数据包的过滤以及防火墙策略配置
* ****netcat****: 用于快速建立TCP/UDP链接,检测网络的连通性
* ****iPerf****: 网络性能测试工具, 用来衡量网络吞吐量/带宽
* ****tc****: Traffic Control, 用于显示/修改网卡配置的工具
* ****curl****: 基于[libcurl](https://curl.haxx.se/libcurl/" \t "http://sniffer.site/2019/01/17/Linux%E4%B8%8B%E5%B8%B8%E7%94%A8%E7%BD%91%E7%BB%9C%E5%B7%A5%E5%85%B7%E4%BD%BF%E7%94%A8%E6%80%BB%E7%BB%93/_blank)的数据传输工具, 支持HTTP/HTTPS/FTP/RTMP/RTSP/SCP等常见协议