

程序调试相关命令

第一步: Jstack [jobid] 查看程序的函数栈 知道程序在哪里卡住

```
ERROR: [vn096097] ##### resend fails, sqn = 84, status = 0x1, opcode = 0x18, retry_cnt = 0
ERROR: [vn096098] ##### resend fails, sqn = 89, status = 0x1, opcode = 0x18, retry_cnt = 0
SWCH Error: MYID = 27573(vn96097, CG4): Got one SEND to 19381(vn94694, CG2) failed (source_guid=0x57761, dest_guid=0x171e7), wrid = 4092, op
NEOUT_1), time = Wed Jul 21 12:34:25 2021, cqn = 51, type = MPIDI_CH3_PKT_EAGER_SEND(0), v = 0x50008f875e98: <swch_drill()@src/mpid/ch3/channe
SWCH Error: MYID = 27579(vn96098, CG4): Got one SEND to 19387(vn94695, CG2) failed (source_guid=0x17763, dest_guid=0x571e7), wrid = 4092, op
NEOUT_1), time = Wed Jul 21 12:34:25 2021, cqn = 54, type = MPIDI_CH3_PKT_EAGER_SEND(0), v = 0x50008f875e98: <swch_drill()@src/mpid/ch3/channe

[swict@sw_hpc_5 15w_atoms_mini_local_index]$ jstack 2731581
-bash: jstack: 未找到命令
[swict@sw_hpc_5 15w_atoms_mini_local_index]$ Jstack 2731581
Process data [235]... 99.97% (81896/81920)
20 seconds no data. Exit!

=====
T0 -> 0-19366,19391-81919
-----
-0- /home/export/online1/mdt00/shisuan/swict/wuyangjun/05_local_index/fhi-aims_MPE_03_local_index_scaling/bin/aims.191127.scalapack.mpi.x
-1- main at aims.f90:38 ([0/81896])
-2- aims_real at aims.f90:45 ([0/81896])
-3- aims at main.f90:178 ([0/81896])
-4- initialize_scf at initialize_scf.f90:290 ([0/81896])
-5- initialize_grid_storage::initialize_grid_storage_p1 at initialize_grid_storage.f90:757 ([0/81896])
-6- synchronize_mpi::sync_workload at synchronize_mpi.f90:1284 ([0/81896])
-7- pmpr_allreduce ([0/81896])
-8- PNPI_Allreduce ([81896/81896] T0 138/v91369 139/v91369 140/v91369...)

=====
miss stack lists
-----
task: 19367-19390
node: 94692-94695
=====
```

程序调试相关命令

第二步：swrrk + 节点号 + 主核号 查看主核上的报错信息中有没有pc值

第三步：swaddr2line -Cfe [程序路径] [PC值] 查看报错具体位置可定位到具体行

```
]$ pwd
shisuan/swict/liurenyu/02_TEST/07_S2D_Test
t]$ swadd2line -Cfe /home/export/online1/mdt00/shisuan/swict/liurenyu/02_TEST/07_S2D_Test/make.sh: 不可识别的文件格式
t]$ swaddr2line -Cfe /home/export/online1/mdt00/shisuan/swict/liurenyu/02_TEST/07_S2D_Test/ntpoly/SMatrixAlgebraModule.f90:86
st]$
```


Roofline model 模型在一个计算平台的限制下 到底能达到多快的浮点计算速度

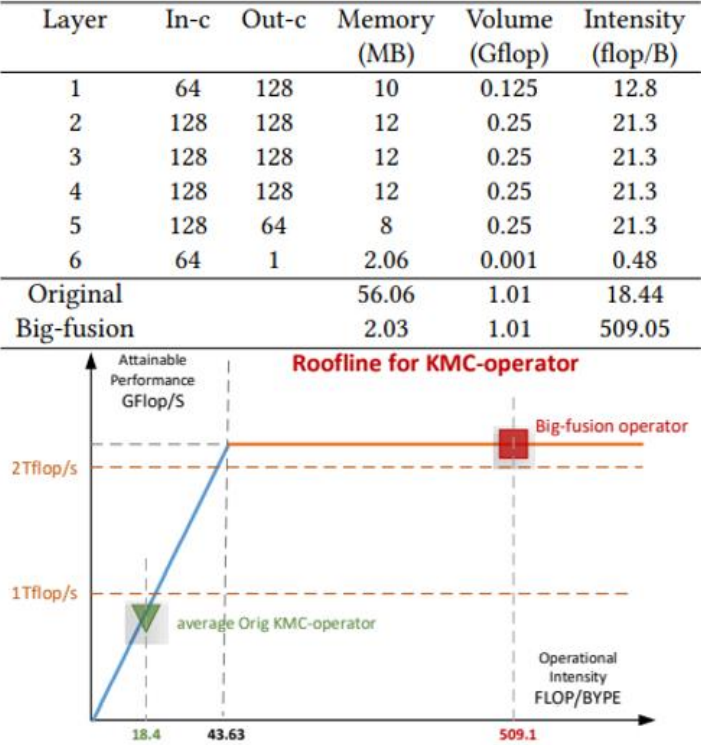


图 9：新神威架构的能量核的 Roofline 分析。Roofline 分析显示计算能力、内存带宽和数据复用之间的相互作用。bigfusion 运算器的算术强度为 509.1F（FLOPS/Byte）。因此，可达到的性能要比原始版本高得多（网络的每一层都被单独调用）。

Roofline model

在真实世界中任何模型（例如VGG/Mobilenet）都必须依赖具体的计算平台（例如CPU/GPU），Roofline model使用Operational Intensity（计算强度）进行定量分析，并给出模型在计算平台上所能达到的理论性能上限公式

1.计算平台的两个指标：算力 π 和带宽 β

- 算力 π ：也称为计算平台的性能上限，指的是一个计算平台倾尽全力每秒钟所能完成的浮点运算数。单位是 FLOPS or FLOP/s。
- 带宽 β ：也即计算平台的带宽上限，指的是一个计算平台倾尽全力每秒所能完成的内存交换量。单位是Byte/s。
- 计算强度上限 I_{max} ：两个指标相除即可得到计算平台的计算强度上限。它描述的是在这个计算平台上，单位内存交换最多用来进行多少次计算。单位是FLOPs/Byte。

$$I_{max} = \frac{\pi}{\beta}$$

Roofline model

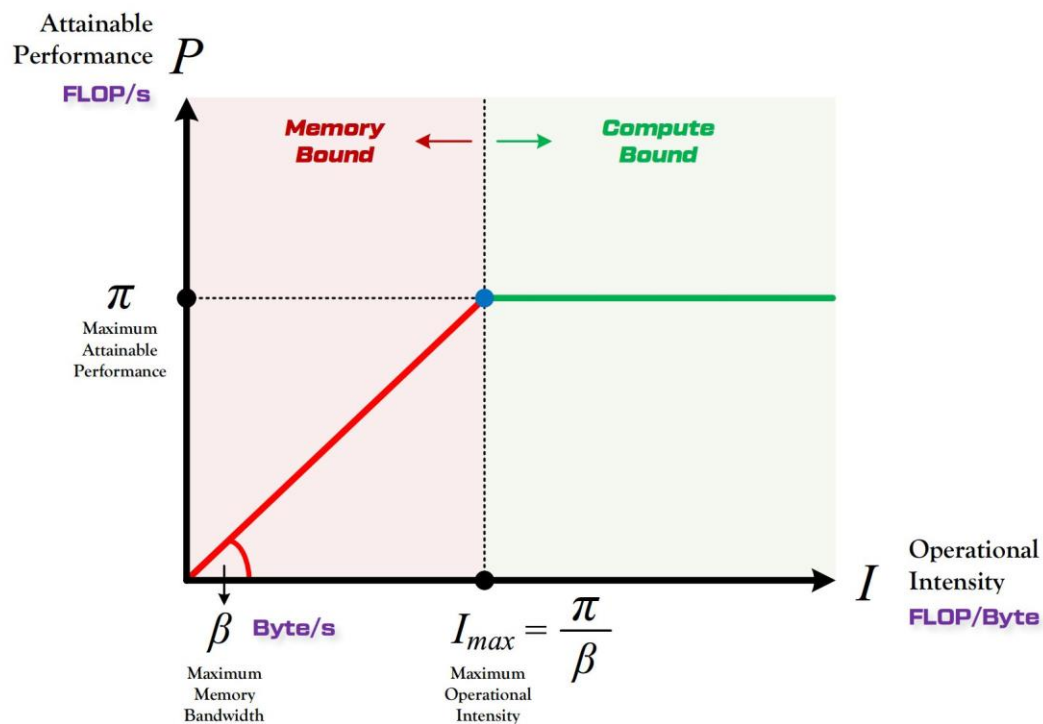
2. 模型的两个指标：计算量 与 访存量

- 计算量：指的是输入单个样本（对于CNN模型来说就是一张图像），模型进行一次完整的前向传播所发生的浮点运算个数，也即模型的时间复杂度。单位是 FLOP or FLOPs。
- 访存量：指的是输入单个样本，模型完成一次前向传播过程中所发生的内存交换总量，也即模型的空间复杂度。在理想情况下（即不考虑片上缓存），模型的访存量就是模型各层权重参数的内存占用（Kernel Mem）与每层所输出的特征图的内存占用（Output Mem）之和。单位是Byte。
- 模型的计算强度 I ：由计算量除以访存量就可以得到模型的计算强度，它表示此模型在计算过程中，每Byte内存交换到底用于进行多少次浮点运算。单位是FLOPs/Byte。计算强度越大，其内存使用效率越高。
- 模型的理论性能 P ：我们最关心的指标，即模型在计算平台上所能达到的每秒浮点运算次数（理论值）。单位是 FLOPS or FLOP/s。

Roofline model

模型在一个计算平台的限制下，到底能达到多快的浮点计算速度。更具体的来说，Roof-line Model 解决的，是“计算量为A且访存量为B的模型在算力为C且带宽为D的计算平台所能达到的理论性能上限E是多少”这个问题。

所谓“Roof-line”，指的就是由计算平台的算力和带宽上限这两个参数所决定的“屋顶”形态，如下图所示。



算力决定“屋顶”的高度（绿色线段）
带宽决定“房檐”的斜率（红色线段）

Roofline model

$$P = \begin{cases} \beta \cdot I, & \text{when } I < I_{max} & \text{Memory Bound} \\ \pi, & \text{when } I \geq I_{max} & \text{Compute Bound} \end{cases}$$

· 计算瓶颈区域 Compute-Bound

不管模型的计算强度 I 有多大，它的理论性能 P 最大只能等于计算平台的算力 π 。当模型的计算强度 I 大于计算平台的计算强度上限 I_{max} 时，模型在当前计算平台处于 Compute-Bound 状态，即模型的理论性能 P 受到计算平台算力 π 的限制，无法与计算强度 I 成正比。但这其实并不是一件坏事，因为从充分利用计算平台算力的角度上看，此时模型已经100%的利用了计算平台的全部算力。可见，计算平台的算力 π 越高，模型进入计算瓶颈区域后的理论性能 P 也就越大。

Roofline model

$$P = \begin{cases} \beta \cdot I, & \text{when } I < I_{max} & \text{Memory Bound} \\ \pi, & \text{when } I \geq I_{max} & \text{Compute Bound} \end{cases}$$

·带宽瓶颈区域 Memory-Bound

当模型的计算强度 I 小于计算平台的计算强度上限 I_{max} 时，由于此时模型位于“房檐”区间，因此模型理论性能 P 的大小完全由计算平台的带宽上限 β （房檐的斜率）以及模型自身的计算强度 I 所决定，因此这时候就称模型处于Memory-Bound 状态。可见，在模型处于带宽瓶颈区间的前提下，计算平台的带宽 越大（房檐越陡），或者模型的计算强度 I 越大，模型的理论性能 P 可呈线性增长。

Roofline model

$$P = \begin{cases} \beta \cdot I, & \text{when } I < I_{max} & \text{Memory Bound} \\ \pi, & \text{when } I \geq I_{max} & \text{Compute Bound} \end{cases}$$

- Roofline 模型讲的是程序在计算平台的算力和带宽这两个指标限制下，所能达到的理论性能上界，而不是实际达到的性能，因为实际计算过程中还有除算力和带宽之外的其他重要因素，它们也会影响模型的实际性能，这是 Roofline Model 未考虑到的。

新学期目标

1.开题

2.小论文

3.评奖学金、六级刷分（上500）