

平台开发的需求和问题

我们目前正在开发平台的数个功能，包括：

分布式图像重建

分布式图像重建主要是要解决大型PET系统重建图像计算量大，速度慢的问题。比如目前想要解决的1.4米的系统的重建，需要30000到50000个CPU小时（3GHz）。图像重建主要是大量矩阵乘法或者类似的计算密集型的操作，我们希望能够开发出分布式重建软件，利用多节点上多GPU来快速获得结果。目前开发的难点包括：

- 节点间的通讯问题

重建采用迭代算法，每次迭代后需要将结果在多节点间合并，然后广播，需要一定的节点间传输带宽。比如对于尺寸为195x195x527的图像，每次迭代就需要所有节点交流并广播100MB*节点数左右的数据，如果使用星形的数据交换方式（所有worker节点传到master节点，融合后广播回到worker节点），在节点数比较多时可能会成为比计算本身更加耗时的瓶颈。如果使用环形的数据交换方式，将数据在worker上分段环形交换融合，可以使得交流的数据量不随节点数增加而增加，但相应的程序开发复杂度会略有提高。

- 软件开发复杂度问题

整个重建算法相对复杂，开发成分布式的后需要解决的问题更多。同时还有一个更重要的问题是重建算法有多种模型，需要保留灵活改变算法的可能性。目前的方式是将计算密集型的部分抽象为基本的算子（主要是矩阵乘法，投影反投影等），采用C/C++/CUDA C开发。上层使用Python开发生成完整计算图的程序，每个节点分配计算图的子图，再辅以一个计算图引擎来计算。目前我们是使用Tensorflow作为计算图引擎并处理分布式交互。但是Tensorflow性能上可能会有限制（比如饱受诟病的GRPC）。

多机并行蒙特卡罗模拟

和图像重建一样，蒙特卡罗模拟同样需要非常大的计算量。但是面临的瓶颈不太一样。

- 分布式文件系统问题

蒙特卡罗算法由于其仿真的每个粒子间相互独立，有着非常好的可并行性，节点间几乎不需要信息交流，可以认为就是多个计算节点分别跑出结果后把输出的文件融合即可。它的难点在于研究中使用的蒙特卡罗模拟为了后续处理，会输出比较多的中间信息，需要一定的储存空间。目前采用的是GlusterFS，暂时可以满足要求。但是期待有更好，更稳定，更易扩展的选择。

后处理问题

不论是图像重建还是蒙特卡罗模拟，都有着后处理的需求，包括：

- 特定的文本处理

典型例子是蒙特卡罗仿真输出的一个或多个CSV文件，需要处理为更加结构化的数据，或者储存为二进制格式等。此时处理的文件大小从若干GB到数TB大小不等。此时处理程序相对简单，但是需要多节点处理和比较大的外存IO（或者能利用内存的文件系统，并有充分的内存）。

- 其他算法（比如机器学习）处理

典型例子是我们需要对大量重建出来的图像使用机器学习的算法进行处理，比如目前正在使用的基于Tensorflow的深度卷积网络处理图像。此时往往需要处理数百GB到数TB的训练集数据，但是和上文的文本处理不同的是它需要的仅是高密度的读取，几乎没有写操作。同时多个节点训练网络的时候，需要同步网络参数，也会有分布式重建中一样的带宽问题。

整体平台构架

对于整套系统，我们计划将以上各种程序以及其他的数据采集和用户相关的一些操作打包成微服务，最后部署到一个分布式系统上，提高各个组件的复用性。可能会有如下的需求和问题：

- 我们的平台面临的并发应该不会很高，但是每次响应的时间会比较长，比如重建或者仿真可能发起请求后需要半小时到数小时之后才会有结果。

- 能够简单，灵活地控制不同服务的工作的节点的数目。不同任务的优先级和算力需求会随时变化，需要能够有比较好地弹性地分配节点，有时会需要将部分已经开始工作的节点停止当前工作，转而优先供给其他任务的可能性。
- 是否有进一步虚拟化的必要。现在的程序均是直接运行在linux系统上的，是否要切换到容器上以获得更好的开发和管理可能性呢，以及如果使用Docker，其性能损失会不会成为问题。

结论

请问面对以上应用需求，我们目前的技术选择是否正确，您觉得是否有更好的选择？对于提到的问题，在您看来是否还是问题，如果是，需要引入怎样的解决方案呢？