微软亚研院 薛继龙

01:01:01

【P2】

其实我做了三四年的compiler，然后分享一点就现在的问题。我讲一下我们的观点，包括问题以及我们的认为的答案是什么。然后我们先稍微讲一下我们DNN compiler stack，大概我是觉得这样分。就是说前端语言、计算抽象、编译方法、硬件抽象，我们从这4个层分。那么这4个层就是说现在普遍的方法就是前端语言，我们其实大部分都在做Python binding。然后计算抽象就是我们以OP为中心的DAG。然后编译方法，我们其实刚刚很多老师都在讲，就说其实有编译是个非常大的搜索空间，我们有各种的方法需要去找到到底是怎么最优的对吧？

然后硬件抽象其实现在我觉得比较缺乏的一个统一的硬件抽象，就会导致现在的硬件适配是非常困难的。那么产生潜在问题，对于上层现在的这种binding的情况，就导致上层的语言那边的编译优化和 AI的编译优化一些分离。然后产生很多overhead，我们觉得op的粒度是为了描述算法用的，它不适合去做硬件mapping，所以这是一个错误的粒度，然后会导致很多潜在问题。然后超大的搜索空间，那么我们优化有大量的编译时间。然后最后就说是难以适配到不同硬件。

【P3】

直接到下一页。下一页是我们最近总结出来应该的一个AI compiler，从我们观点觉得应该是什么样子的。

上层的话，这样其实今天上午也讨论到了，我们认为就是说未来DNN编译应该是作为语言的特性就直接嵌入进去，然后但是不一定是走Julia这条路线，因为我记得谁提到一个生态问题对吧？然后我就先不说了，然后时间有限。

第二层我觉得更重要，下面三层就是说计算抽象，就是说到底 DNN的计算有什么特性对吧？就说他跟作为我们做DNN compiler到底和其他compiler有什么区别？我们认为是有计算抽象。计算抽象，刚刚老师一直都在说编译是一个搜索空间很大的问题。那么怎么去解决问题，我们认为不是把它变成搜索问题，而是把它看成一个调度问题。有一个形象的例子，要做原来做memory reduce【01:03:00】的时候，在一个非常大的cluster里面去调度的时候，为什么呢没有这么多搜索问题呢？然后我们现在也是想从这个角度考虑，就说我们事实上把一个op，或者一个图调度到GPU上面也是这样。下面很多的基本单元，我现在想map上去，如果看成调度问题这个问题就会简单很多了。

然后这个工作是我们去年Rammer做的，其实是在做计算抽象。我们试图把计算打成一堆小的task，然后去做调度。那么这个task怎么做还是很困难，所以我们现在在做的是Rammer的第二版本，然后一个新的。我们本来现在投Soft【01:05:00】，还是有点不ready的，我们准备投今年的OSDI。然后稍微分享一下，就是说事实上我们需要一个下面的硬件抽象。硬件抽象，我们现在认为就是说，给所有的硬件有一个统一的软件抽象。就它带有核心的可能有两个东西，一个是叫做dynamic tensor primitives。。。【01：04:00缺失】就是说一些基本的pipeline model。然后有了硬件和软件的抽象之后，就说相当于我们的所有的算子都可以有非常小的这种primitive的组成之后，上面的一个好处就是说编译就变成一个白盒的编译了，就是说我可以通过组积木一样组起来，就是把一个执行图变成一个schedule【01：05:00】 plan。这样做完之后，比如说我的编译首先不需要去搜索了，比如说我可以做one shot的编译，然后其次就是说整个对上面的调度的空间也会化简很多。然后也就稍微回答一下上午敏捷的问题，就是说我们试图这样做完之后，就说你上面的表达性不管表达什么样子的，下面可以直接就是一键的给你映射到GPU上面去，然后这样去做到了和TVM和现在的CUDNN之间的一个比较好的trade off，所以我只是抛这么多观点，然后最后细节我们现在还在讨论。