### 20210709\_094739.m4a

说话人1 00:00  
目前的优化是大部分还是靠工程师的，那么有没有更加自动化的优化方法？那么我这里举几个例子，一个就是在cv m里面它有 automatic的优化的功能，我觉得也是一个未来的方向。第二个说到我们在LP里面，大家都现在在CV里面也会比较火的这种预训练模型，那么这个预训练模型我们现在可以看到，从早期的比如说几亿级别的，或者再找几个更小几百万参数级别的，一直到比如说 But b类级别的，然后一直到现在的，比如GDP三是千亿级别，一直到我们现在比如说，国内像华为、阿里。

说话人2 00:52  
百度都推出万亿级别的模型。

说话人1 00:55  
所以参数增长规模是非常大的。

那么这里就带来几个问题，就是一个问，一个问题就是说我们这些外机的模型，它带来的收益首先就是说非常小，然后另外它的成本非常高，就算一次的成本非常高，所以它的运算其实或者它的模式就不能采取以往的预约模式，就可以串加翻清热的模式。就是说因为你翻QQ的成本都非常高，所以我们最好能够想一个办法，能够不让他翻75年就能够在下一个任务上去执行，这个是目前我们也在关注的一些问题。

对于所以我这里举了两个例子，一个就是说我们在预训练的时候，比如说我用拍套去，那么我通常会选择在做大规模训练的时候，我们通常会选择两两个包来去加速它，一个就是。

说话人3 02:08  
 DSP的。

说话人1 02:09  
它就是在相当于一个深度优化的一个拍到这个插件，你就可以用它，因为它提供了很多高级的蒜子，这些蒜子相当于比如说你在开发区里面去实现一个传输的，里面会有非常多的算是他把一些几个可能复杂的算式组合就给优化一下，然后这样在后端可以运行的速度更快。

说话人4 02:45  
传播的话也。

说话人1 02:46  
是这样的话，它可以把这个速度可以上去，然后另外它还可以做一些自动的这个就是并行，然后比如说像数据并行，还有模型并行，还有流水线并行这样的方式，这样的话它的效率会非常高。

对，然后它还有一个特点就是说它可以针对你在influence的时候就推断的时候，它对不同的下游的任务也可以做到优化，这样会我训练出一个模型之后，还可以进一步在不同的下游任务上也能够得到非常好的性能。

说话人5 03:27  
非常生产的一个条件。

说话人1 03:31  
这就是它的比如说我们这里列了一个例子就是说，在DB识别里面的一个核心的东西就是DB分子，就这里面我们可以看到有很多的算子，其实有些算子是可以把它融合到一块的，然后这样他比如说我下面直接对应某个扩大的非常高效的实现，或者一非常底层的某些硬件的实现，它的一个比如说硬件的运算就可以对应这里的一个组合，所以它要在这个层面进行一个标准，就算是融合。

这个是在另外一个我们经常用的大规模训练的框架，就是 MAC上面它也是用这个算是融合的方式就是说，我们可以看到上面是一个常规的操作，可能需要几步，然后对于比如说从里面他就带了很多的如何算值，这样我们在算的时候它就自动把上面这个放假了，上面这些计算图就优化下面这就是那么还有一个经常用的 Top script，它是一个非常简单的想法就是说，因为套酒本身是一个动态语言，所以他在 influence的时候，如果你还采用这种动态语言的话，它就效率会比较低，所以它就相当于用 jit的形式。

说话人6 05:09  
然后把它完成静态图。

说话人7 05:12  
然后。

说话人1 05:13  
这样的话可以在做推断的时候就可以效率更加高。

对，最后总结一下说，我觉得就三方面的内容，就是一个现在的大规模的训练模型，它对训练和推断的效率要求非常高，当然对于容量要求也非常高，但是最关键的很多效率，现在说我训练一个大模型，它消耗的时间太长，就如你怎么样去加快，那么包括我如果做一些后端的优化，然后另外一个我觉得一个比较重要的就是不同框架下面的模型沟通问题，比如说最好有一些自动的模型转换，我们甚至也在做一些这种工作，这个工作是什么？

比如说因为我本身是做这样处理的，那么我们之前做了一些工作，这个工作就是说我们可以根据不同框架的 API文档就读他的API文档，知道哪个API是干什么的，然后实现不同的语言之间，然后可以可以进行相互的转换，大概我们解决这个方法的一个思路。

然后另外一个就是训练和推断的之间的需求差异，在深入学习里面其实有一个非常重要的问题，就是训练和推断是割裂的，比如说我我，很多时候你在训练的时候，可能比如说精度要求不是很高，然后在推断的时候相对来讲精度要求会高一点。

然后另外说现在比如特别对于大模型之后，训练的阶段基本上都是分布式的，所以训练阶段的代码和在测试阶段的代码其实是长得非常不一样的。

你怎么比如说能够像比如说一份比较干净的或者一份推断的代码，能够自动把它给它扩展到以它为核心，然后扩展到训练非常轻易的点，就是非常容易的把它给扩展到训练阶段的，代码我觉得也是非常重要的。好我的报告就到这里，谢谢大家。

说话人8 08:01  
非常感谢基本的介绍，但其实总结得非常好，第一个问题跟刚才李老师也提到了，也要接近一个大模型的问题，此后他还提到了模型之间的转换，怎么样通过编辑能更自动的转换，希望编译器的能够我们帮算法做算法的专家们做这些事情，包括还有训练和推断代码怎么样能够比较自动的转换到一个训练的，并且部署到不同的平台上去。

那么在听完了两位算法专家的对编辑的诉求，我们就找一个做编辑的专家来。

说话人9 08:42  
 echo一下对吧？我们有请计算所的。

说话人8 08:47  
周慧敏老师也是我们编译。

说话人10 08:50  
周边系里面的平台来给大家。

好，其他影响。老师昨天孩子有的。

说话人11 09:02  
这个吗？

说话人3 09:02  
这个不可能的培养不同。

说话人12 09:07  
各位大家好，非常荣幸，其实因为我们这个团队一直是在做传统的，所以 Ei出来的时候，我们最早跟寒武纪陈老师我们一块聊，当时他们刚开始做AI的时候，其实大家就是说因为他们做芯片，然后当时就是说我们就有过一次讨论，就是说我们到底需不需要一层类似cos这样的一层语言，当时算是还相对蛮固定的，但是当时后来我们讨论的结论还是就像林老师提到，还是让大家不要受硬件约束，要能够想玩什么玩什么，所以我们觉得寒武纪也是需要这样一个一层比较粘的严的，所以后来我们就做了bnc，你可以看到不管怎么样，我们当时的思路都是从一层摘录的角度来出发，因为我们没有AA的背景，我们没有 ei的这些知识，所以我们都是比较平的去看这个代码的，但我们在这个过程中做的这几年过程中也会发现，我们用传统编辑的思路来优化，确实有一些特殊的机会，所以我今天想抛出来一个问题就是说我们到底传统电影它们之间结合点在哪里？

所以这是我们团队之前的一直做过的事情，最早就在之前的大概其实我们从五六十年代开始做大型这边团队一直就存在，然后到从2000年以后，我们最早是给intel做orc完全就是纯面向一个芯片架构的编译，后来开始我们所做龙龙芯，我们也开始面向龙芯做编译，然后又当时声威开始启动我们，又面向声威，就帮声威做编译，然后一直到后来现在开始给寒武纪做编译，所以一直是处在一个通用编译器的这样一个层次上。

然后我们这几年有一些思考，当然我们刚才几位老师都提到了，做恋爱的话一个算子nei的，一个就是svo子此间的对吧？然后算子的话，这个是我们当时做bnc语言的一个很很主要的一个诉求，就是把当时编译器的基本上的一些像modern dream，compiler的analysis和optimization都加进来，然后这些常规的一些优化是可以做的，这一项扩大做的这些事情一样。

然后 auto tune的一些东西，这些传统分开了，做o to o就是你的一些技术要加进来，然后影像的镜头品牌精神是我们缺失的，然后但是现在含5g的人在往里做，但这一块因为我们的background比较弱，我们就没有价值了。

但是后来我们做了很什么的一块，就是我们做了ain，semantic的Ollie就是做AI的人，其实AA你们从你们的角度，你们对 svo是干什么的，然后他每一行代码干什么其实是非常清楚的，但是反映到compiler从代码的层次来看，它其实不知道这个事情的，然后这样就miss掉一些东西，然后我们从算起的角度把这个知识再加进来，然后这样会有一些特殊的优化机会，所以这个是在自定义算子里面去做优化的时候我们的一些想法。

然后其实后来我们就发现做AI的人非常care做 operator fusion，就是因为老师提到的可能让起的代价实在太大了，然后尤其是现在有非常多的这种长尾算子出来以后，其实一个做可能比如是 operator fusion，然后另外一种就是像rem这样的，他可以把好多小的课同时的浪起来对吧？然后这样去做。

但是这个东西我们其实两一年多前开始做这个问题的时候，我们就说为什么要专门去做operator的fusion，如果把所有代码我如果看到的不是一个svo层面的，也在不是一个算字层面层那个代码，然后我从图的层面看的话，我也不是只看到no的，我把图和代码结合到一块儿，我不就把原来大的程序返回出来了，我拿到大的程序对传统的本派的人来说，非常经典的look对吧？但是在这里面当然有非常多的问题，你怎么把这个东西feels到一块，然后你怎么把这个东西用reverse回去，把这些所有的往裤带上丢的，我们拿裤带来说哈嗯，裤带上丢的这些所有的饮食的循环，你再把它打开，然后你要把所有的memory给它搞好，其实是一个繁琐的过程。

所以我们现在大概两年这两年的时间，我们bill的一个就是能把house和科隆feels到一块去做哪个CS的这样一个框架。

我们这个框架我们现在是其实当时我们咖啡的是出去的paper是去用oneself来做的，然后因为我们当时是其实因为关起门来说，我记得棒很像酷的，矛盾是差不多的，然后高分Cr我们也用在了深a上，然后当然它support的平台非常多，当然open cell不是一个很好的推广的语言，但是按照我们做这个呃一些前期的验证是可以的。

就是说我们是把host代码和QQ代码拿到一块，然后把他们的ASP层面合到一块，然后在这上面去一块去做，偶尔留人印到lvm的l上面去。然后这样带来的一个问题，您feel到一块以后control flow analysis怎么做对吧？

然后在我们的papier里面解决了。然后data flow的拿起怎么做对吧？我们也解决了，然后inter procedure的analysis怎么做我们也解决了，所以我们把把基础的base的analysis都解决了，然后现在我们在做在这上面在做怎么做field，简单抛一个例子来看一下，我们这样子fusion和传统在operator层次来做filth有什么区别？这个是 Lstm的例子，然而也是拿它来做一个很好的JavaScript，然后你可以看到它的大量的模式，你如果把它循环展开以后，然后你去做传统的底盘政策，拿s你会发到发现它的循环的依赖是这种wave front的模式，我们自然就可以用传统的 looks giving这些东西来找到这样一个排来的模式，然后这样的话我就知道我沿着维护上去的方式是可以去做并行的，对吧？

这种情况下我可以去做费用审，我知道每个客人都比较小，我从客人的层次，我从每一个酸子的层次，我知道他可以去做费用是我可以把他们费用造一块，费用造成一块的好处是什么？

然后你可以做得看就是use，这个其实是传统以前大家做两种就是不会搞定的， RAM这种方式，它可以把多个kernel一块loss起来，但是它不会解决多个kernel之间对它有优势的问题。

传统的operator fusion是在解决data reduce的问题，但是他不会考虑injure因特尔克诺的paragraph的问题，所以但这个东西因为赵杰老师也在，赵杰老师就更擅长了对吧？就这个东西从compared的角度，它是天然能够因为它有很多的model quality jue的，然后传统的oo的model都是可以同时再兼顾parlor them和老k的，所以这就把原来大家做的一个比较面向AA做的一个model，然后其实从pallor的角度你脚猫都好，其实它是一个传统材料可以解决很好的问题。

所以我们的角度是从当然我觉得这样也有问题，因为你非常费事，很多AI的 model的信息其实从就是语义然后analysis的角度， power的defense，然后data race这些东西其实都丢掉了，对我来说也是机器人不好的，所以我们现在在写作文就遇到一个问题就是说我们到底这个东西这个界限到底该怎么定，所以这也是我们一直 paper没有organize好，就觉得这个界限我们自己都还没有搞清楚。

所以今天在座的各位专家在这也是希望能够我们在这个问题上帮我们往前走，进一步的一个回答。好，然后我的学校就是这样子，谢谢大家。

说话人13 16:56  
好。

说话人8 16:57  
谢谢崔老师的分享，其实是个非常好的大的问题，编辑和VIP到底边界在哪里？好，我们紧接着到再下面一层编辑在上面就是芯片架构了，我们这边走着两位非常厉害的芯片架构师，我们先有请清华大学的尹守一教授，他们 think芯片系列也是明道中外对吧？

所以所以我很早就跟李老师交流的时候，他就给我吐槽过，当时用TM做因为基础去做编译的时候遇到了很多坑，反映了很多问题，所以今天我们就从芯片架构的角度来看看编译器到底看过什么坑，或者我们对于对我们现在在场这些做编译的专家有什么样的一个一个建议或者诉求，很欢迎。

说话人6 17:54  
大家好。

说话人2 17:56  
其实没有太多跟奖励吐槽TV，因为 Tv呢是。

说话人6 18:02  
在旅游的工作。

说话人2 18:05  
然后我们一直觉得是我们没用好而已。

今天想跟大家探讨一下讲编译器，因为我其实做了挺多年的建议，就是本身讲了一说我是这个架构师，但是其实如果细数的话，可能做编译的时间还更长一点，因为之前一直在做可重复计算，就是这个粗力度的可重构的架构，所以其实那里边架构本身的问题好像其实倒不太多，因为它比较规整，更多的问题是怎么把这个任务给埋进上去，所以做了很多年。

我刚才听了几位专家的前面的分享以后，我跟崔老师的有一个应该讲是共鸣，就是说其实我们以前听大家Python讲过一句话，他说在现在 Computers开始沟通内置有一些 own seniors病，我自己觉得就是说iPad里面有很多的things，然后现在很变得很烦c比如说c超了100倍，其实编译里面也有相当多的，比如说大家刚才讲到的，其实很多前面专家讲到了 future的问题，陈老师也讲到，其实确实那里面非常典型的那种风险，很多经典的方法抛一些肉什么这些，包括不用抛一些肉，这么翻碎的东西其实都能解决的挺好。

但是当我们把它放到 AI的背景里头以后，搞出来的图什么的，它变得很fancy，大家觉得这是一些新东西，但是如果去看里边的话，可能会发现一些相似的东西，但是这话说的怎么不好，因为今天我们在AI大会的现场，把他说成一个很旧的东西，好像有点砸场子的现象。

我想那就我这里边总结了几点，就是我们在做芯片的过程里面，然后要做编译的时候遇到的一些问题，这些问题有一些是我想可能相对来讲是比较底层或者比较微观的一些问题。

首先前面几位嘉宾讲过，其实这个DNA它肯定是一个多级优化的问题，至少我们从现在整体的流程上看，首先从我们的框架出来，然后大家都会把它表达成这段图，然后如果把这段图。

说话人13 20:32  
每个节点再拆开的话，典型的里面的计算其实就是循环，因为。

说话人2 20:37  
不管是转机还是什么的，都是几层循环在里面，那就变成了一个很经典的一个循环的映射或者循环本身定期优化的问题。

再结合到底层的硬件架构的话，因为底层架构上面它有大家基本现在都是一个阵列的计算的架构，阵列之间阵列里边的单元之间有数据流，然后这列又有几级的存储，这里面就数据跟计算之间怎么去买进买进的，过程里面怎么再去考虑仿存优化的问题。

说话人14 21:13  
实际上就变得很关键了。

说话人2 21:15  
我自己的看法就是说其实如果讲说现在 I的边有什么特别之处，我觉得最大的特别之处，它似乎应该是变成一个以存储优化为中心的一种方法，过去我们去做。

说话人6 21:29  
比如说这种病毒用色的时候。

说话人2 21:32  
更多的会考虑钻石怎么定制上去实现功能，或者说我们去考虑怎么去展开，或者说去做move的传送nation，去把更多的病情录给提取出来。

但其实在 AI的计算里面，不管是卷积还是全连接，或者是传统的这种，反正它基本都是张亮或者矩阵它的定型度其实是很清晰的，但是这里边它就会有很多数据的 low的store use这些问题，而且尤其是很多我们这种张量计算，它其实从数学上来讲就有不同的计算的方法，比如说矩阵，我们可以用内机的方式做。

说话人14 22:11  
也可以用机器的方式做。

说话人2 22:12  
这种不同的计算的方法，它就带来的影响是说你是计算量大了还是放存量大，或者这里边会有一个隧道。

所以我觉得从这个角度来讲，编译它应该更多的考虑存储或者说以存储优化为中心，更它的优先级可能甚至更高于说我们去怎么去做计算的飞行。

这里面就简单的看大家现在把它编把整个编译过程分成了前端，终端、后端这三个层次，前端上来讲基本完成，就是说我们把一个框架给出的模型，我们把它抽象成 ir的方式。

这里边我们自己感觉现在的问题本身现在是一个本身工具链的一个江湖草莽的时代，所以 l很多到底怎么去形成一个统一的事实上的标准，其实是个问题，要不很多转来转去的话，这些工具尤其是很多是开源的工具，那么它在剧烈的变化，所以这里边会给应用和开发带来很多的不确定性，还有很多的工作量。

第二个部分我们叫终端，这里边其实大量的优化是在这儿，比如说刚才提到的计算图上的优化，包括如果把图再往细节一步看的话，张亮算子的。

说话人15 23:44  
到硬件资源。

说话人2 23:45  
的优化，刚才说的存储数据流这些，其实它应该是在层次上然后后端更多的是一个从。

说话人15 23:54  
然而到最。

说话人2 23:55  
起码的一个翻译的过程，我想这个过程其实跟标准经典的电影其实是差不多的。

我这里面想有几个细节的问题想跟大家分享一下，太完整的，我想宏观的东西大家都很专业了，我就讲几个我们比较有感受的具体的问题。第一个问题就是刚才很多跟大家提到的算的优化，其实所谓的算的优化我自己理解，其实它实际上这个概念里边有两层意思，一层比如说我们常见的说像com5bn这些操作，就这些算题，我们会很自然的把它给 Q起来，因为它本身就数据从数据应用上来讲就带来好处。

另一层前面也有跟大家提到，就是说我们真正的把不同的内容给提供起来，那么这样的话解决一个问题，就是不同的层之间上一层的输出变成下一层的输入，实际上就会带来很大量的低端的访问。因为从架构角度来讲，片上来虽然再多也不可能把一层的出全都能够完整的实现，所以还会有大部分的数据会需要跑到这个地方去，这种仿真会带来很大的代价。

所以就会考虑我们去把真正把不同的内容给默认起来q起来，但其实 q是就和我们说，如果我们从传统编辑角度看来看的话，就跟这个路就就有点接近了，这里边就会有一个渠道，比如说fields的话。

说话人13 25:34  
单从整体角度来讲，有很多产品。

说话人2 25:38  
会带来的存储访问的减少越大，效率越高。

但是 build有一个代价，就是它实际上本质上是用计算换了存储换个方式，因为从这个图上可以看出来就是说face本身的原理就是说，我把后面一层的某一些市面上的一些点，它的计算过程也全展开了，过去是我们先算一层，然后这些输出变成下一层的输入，但我在这个以后就相当于这一个点，我一次性的去做了好几层的计算，实际上从某个角度来讲，是拿计算化的存储，这时候会有很大的一个就是计算的开销，推的奥数就变得很重要，这是一个问题，说怎么去做配套。

第二个问题就是这个标准的话，本身它是一个很大的搜索空间，我们如果十几层的几十层的网络还可以去搜索，但是现在网络越来越深，1000多层的这种网络或者是超大模型怎么去做全局的搜索。

说话人15 26:46  
其实是很大的问题。

说话人2 26:47  
或者说有什么办法有什么算法能去找一个相对好的 face的解会是一个困难的问题。

这是关于战略融合的就是几个问题。还有一个问题，我们现在网络模型尤其是那次出来以后，它变得非常不规则嗯，原来我们模型直线型前面的输出就是后面的输入，但是变成last以后数据之间的传递关系又有很多交叉了，这时候怎么去实际上也会带来新的挑战。

第二个在图方面的问题，我们觉得是一个在分布式的这种体验能推理一下算子及定型的问题，尤其是现在模型变大了以后。我们会需要有多个独立的硬件芯片来去做同一个模型的计算，比如说像 GPS1万块钱卡怎么去在这个层次上去做算子级的并行化，这也会是一个也会是原型的问题，前面是就是图层面的，再图再往再往下看这个细节一点的话就到了张亮层面了。

那张亮的话实际上它本身。

说话人8 28:10  
这是一个。

说话人2 28:11  
比较典型的循环变化的问题。从硬件的角度来讲，或者是从体结构角度来讲，现在我们通常会有4级缓存偏外的低院，然后偏上有可能58分，还有一些logo版本，再有每个单元里边还有一些集成器，这几级缓存里面，我们去做循环变换的时候，循环变换肯定会影响循环间的数据复用，怎么去结合不同的层次上的存储器，它的访存的速度和以及它容量上的不同的特点去优化或者说去引导循环变换，我们觉得会是个问题。

包括像在台里，包括 Audience的过程里面，它也会是一个很大的。

庞大的一个空间。

其实在这个循环变化层面上，我们本身也有一些疑问就是困惑，比如说像CV I他提出说我用机器学习的方式，我用 AI或者强化学习的方式，我去做循环的变换，但传统的方法上来讲，比如说一个算法或者说或者这种model都是很好的一些业务的方法，我们其实一直在思考，比如说这两个东西到最后它会不会是真正能形成一个等价的结果就是说如果我们从 build或者 transformation的角度去看，一个fields的话，就是张亮见的fields的话，和我们从 AI的计算图的角度去看，这就是这个图上的节点的话，最后会不会是一个统一的最优的结果？

其实我们一直在思考的问题，最后在目前还没有什么答案。当然这里边有一些刚才提到的这些传统的方法，我觉得其实都是能挺好的解决现在的目前的一些问题，值得我们去翻一翻以前的这些论文。那么在张亮算的营销的过程里面就也会有一个新的问题，大家现在从AA计算的角度来讲，这个空间架构就是18开始现在变得很受关注。相比来讲，因为我们张量计算过程里边的数据流，它本身的这种数据通用上的一些特点，使得这种微软开始大家现在很追捧它。

在这里会带来一个问题就是说此阶段开始出来以后，它这个架构本身上面的数据流它是有规则的，然后同时当然因为我们的计算本身还是从实际上来讲，白雷尔是一个实际的计算过程，那就是在时域和空域这两个地方会产生一个交叉，这个问题怎么反映到我们 Png数据流调的模式？

因为它其实是在数据流的层次，会比我们过去看的，比如说数据库用或者是一些代码变换的层次更高一些。

那怎么把让编辑有一个相对宏观的视角，他能去比较好的去调度和选择这个数据，如果他们不能在比较高的抽象层次上去优化问题的话，依靠底层的便利的话，空间太大了，实际上是有编译不能承受的，就是加上这里映射方面的一些问题，还有就是缓存分配的问题。

那我前面讲到就是说我们如果这个直线型的网络，它实际上有明显的输入输出和输入关系。

生产的消费这样的关系，所以我们比较简单的用，比如说大部分8分的这些技术在架构上其实是可以比较好的去服用数据的，但是那次出来以后数据的就是节点间的连接关系变复杂，很多时候就不用说那次就是说比如说像瑞斯耐这种网络，它也是前面的数据会后面好多层以后又用到，这时候前面这些数据怎么办呢我是缓存在偏上，我还是弄到偏爱去，其实也是一些很复杂的隧道的问题。

那这些，渠道其实因为这里就是说编辑器怎么能够它能够在短时间内快速的收到一个好的解，其实是一个挑战。

还有从系统的角度来看，我们现在比如说这种非遗司的这些存储加上迪拜，现在我们的系统基本都是这样，比如说smc配上dy这种系统，那么大家在存储之间有不同的成本，容量肯定pm会大，但是速度 pm m会慢。

在这种情况底下，我刚才讲到就是说这些数据它一定不可能永远都存在，比如说 sm m或者是 pmm怎么在这些不同的这种层次上去最好的去解决一个这些数据的 moment的问题。

所以大家可能看出来，我其实讲这几个问题里边核心的多数还是都是跟这个数据移动有关的，或者说都是跟房子有关，这是我们看现在在 a的言语里面特别值得注意的问题。换句话讲，大家现在看到很多工作它能产生一些好的总体的优化效果，那么更大的程度上其实也是来自于对整个访存这些的优化或者好的，我就讲这些一些体会，谢谢。

说话人8 34:01  
大家谢谢首席带来的分享。

说话人2 34:08  
我们下一位。

说话人8 34:10  
请必然科技的。

说话人2 34:13  
唐山变成科技的研究院的陈院长，唐山来国家从架构的角度也带来一个分享。

说话人5 34:23  
好的好的。

说话人2 34:25  
好，大家欢迎。

说话人9 34:31  
我不是做编译的，我对编辑不懂。

说话人1 34:35  
我只是说这。

说话人6 34:36  
之前做我这三个处理器就愿意愿意这个有点了解。

说话人9 34:42  
我在想反正我是作为芯片架构的架构，我还可以各式各样的东西，加工能玩什么东西，我是比较了解的，所以我是说现在市面上也有各式各样的架构，其实不是我的size，红字是我加了点，我市面上更是大的架构，这就是我们所谓架构黄金时代，虽然新东西不多，大部分都是老东西，我或者说可能全部都是之前有过的，但是提出个问题，就是这么多各式各样的架构或者架构上的选择，对编译器或者上层的框架，我只是提问题的，不说架构好快，我只是说现实情况，下有这些问题我刚才也想了想我怎么提这个问题，后来一想有个现成的，就是可是在他前两天我跟小唐大哥是开篇，就讲为什么对康凯乐有更大的需求或者挑战。

第一刚开始就在讲现在这个架构上的硬件架构上各式各样的情况，我稍微加了点，在技术上加了点红字的，就是大家可以看看现在各式各样的就是不同的，不一定是纯AI的加速各式各样的现在的这种各种这种加速芯片，或者大家叫 bsa或者是传统的架构和dsa的组合可能有多少种变化。

从一个基本的 call的来讲，传统的 style，然后到圣地vita，不管是数据并行指令并行的各种模式现在都有，然后包括加上t字call这种三维的计算，然后往这靠大家从一个大炮到1000多个或者到后面35 25，我不记得多少，我可能后面有个图案到这种班那口，然后后面还有气态的组合，对吧？

然后去PAD再往上就是 market，package就是说一个板子要多多个去对吧？然后再往上那就是多个板卡，甚至多个，我把这些东西放在一起来去运行 are的。我know的，当然这种不同的层次之间对吧？它一个重要的区分就是说它不同层次相应的communication的代价是不太一样的，比如说on t的communication对吧？其他的communication再往 packet之间的这种可能开始，然后在网上对吧鲍汁之间等等不一样。

然后第二个说memory，对吧？以前可能有很多大家用catch用什么这些就是硬件来控制或者调度的memory，其实对compiler或者对上面编程者其实是比较友好，的但是看不到它具体的这些调度现在绝对把 Memory system可能就暴露给compiler，或者暴露给编程的人来用，这是很常见的。

现在很多这种a加速里面的这种sm直接是是一个以静态分布或者需要软件来分配的一个memory，对吧？这个问题复杂性又增大了。

然后第二个问题，其实这种易购的组合也是各式各样的组合，对吧？我传统的CPU编程可能比较容易，但是现在CEO这内存也很多，对吧？然后你现在除了传统CTO，现在sy除了有这个也在讲wyk的extension也在讲，还有DSP的一个层层等等这些东西，这个可能还是最耦合最紧的，相对来说简单一点。然后性t加增色call的这种组合对吧？然后CPU加mto这个组合排列起来就非常多了，你大家是希望说康达能把它都用好，或者编程者都帮他用好，对吧？这是大家的期待。

然后后面还有各种各样特殊的这种架构，就是一些，特殊硬件架构，比如 Cv i对吧？李老师他们做了很多年，包括现在也有一些很关注度很高的这种stuff来做这种CD类的架构，未来可能还有 In memory的computing processing这种东西他们怎么办？对吧我从旁边的角度来去看他们怎么办，而且他还有可能是和传统的这种处理器结合在一起的一个这种东西，对吧？

然后再就是说这些问题的组合就会就会非常多，就是说给我们下面的包括三第三个他提的就是说你拿了一个IP，还有一个问题就是你拿了一个，现在比如说你拿了一个 IP化的这种加速器，其实它是有很多配置的，你是放2个炮，放4个炮，对吧？你每个配sm配多配多少等等，慢慢还可以是几层对吧？我们其实有时候会简化这个问题，可能慢慢还可能比较简单对吧？

但实际上真正在里头marry还要QQ特别复杂的芯片是非常非常多的，现在这些东西。

他就讲的是normal people怎么去变成这个东西，他说你是不是特别聪明的人，可能还行对吧？可能这种复杂度你还能写出几个好的坑，或者还是能写点东西，但是一般大家想想就头疼了，就这种东西对吧？

但是其实我觉得现在大家如果看各式各样的这种新的这种芯片或者什么，这个是很普遍的，而且还有一些说法或者想法，他干脆他的想法是说我干脆把所有硬件里去control的部分都拿掉，都给计算的部分，然后把 Ctrl的难度都给软件，当然我们说不评价好坏，反正是也有这个说法。

好好下一页。对这个。

说话人1 41:16  
给我我是不是下个月我。

说话人9 41:18  
中间还有一页吗？

说话人4 41:21  
这是空白的。

说话人1 41:23  
抱歉。

说话人9 41:28  
对我中间是有一页之前飞速和他讲了一下，讲了一个东西，就是说什么东西是其实很简单，就是什么东西是编程友好的，硬件什么东西是编程会有好的，硬件，就是左边左边是一列编程有好的，右边不有好的，比如说 Memory，你像开始这种可控制的这种肯定就变成友好了，你这种sm这种你直接必须自己去分配的，这肯定不会好对吧？

包括通讯上这个呃，包括。你包括处理器的核心这块。

说话人1 42:03  
我抱歉我怎么。

说话人9 42:08  
没什么反应的，把没有没什么事，然后我举这个例子，其实斯尔巴斯他很厉害，芯片就是说整个一个微分 scale的芯片非常大的一个芯片，然后大家都很佩服他这个工程能力能做出什么东西。

啊但是其实从各个层面来讲，它是编程很不友好的一个芯片，就是说它的存储它每个最小的太阳都是这样的，都是这么一个东西，其实整体来看非常规则对吧？

它的每一个态度里头慢慢你要自己控制，然后 control这个东西。

说话人6 42:49  
你要自己去。

说话人9 42:50  
去控制，当时我不记不太记得他指令其实是多复杂了，但是反正这块你要自己控制，而且很重要的是他每个tell之间他有 Walter router，就是东南西北4个方向就是说他所谓对他flow的流程，你必须事先排，好我 tab里头怎么用？

就是这么多个太好，像这种money号的编程也很难，就是说你一个这一个问题一一个窝扣子迈进到上面，你得排得非常好去。你不仅要把 sm用好，你不仅要把扣子用好，你还要把 rap编程编好，然后你才能达到一定的效果。

所以当然下面还有个paper，他是去怎么去卖品一个科学计算的应用在上面，它可能能实现一个大概百分之四五十的这种效率，这些我觉得他能拿出来的肯定是已经很好了，但是这种编程我们可以想象一下，就是说你把这么多的盒，然后每一个盒的这种配合能配合好，这个是非常难的，但是这种架构所以我其实在下面一页可能对在下面一页其实。

说话人6 44:08  
其实就是说。

说话人9 44:09  
从person那儿，他当然是说我现在有这么多硬件的这种问题挑战，他说反正我们编译器去尽量解决，但是他也给了答案， Mir加sorry这两个项目我们不评价了，在座都是专家，我们也可以讨论，没有答案我没有答案，我只是把前面这个问题摆出来，我其实很关心什么，我作为硬件架构师我很关心什么呢？

还是软件划分的问题，我我们做硬件架构其实一般来说第一部分就是考虑软件划分，就是谁在硬件做，谁在软件做，什么东西在硬件做合适，什么东西在软件做合适，所以最好是个整体效率最高的，但是大家现在做很多架构选择，有很多隧道是吧？

有的时候有的时候想一些很漂亮的说法，其实它背后可能是是有些其他原因对吧？他设计出这样的一个架构，但是我还是希望大家能讨论出一个相对来说，比如一个好一点的划分对吧？

有些东西我的硬件都可能很简单，我硬件增加一些 POS来去做，可能就减少了很多编译或者投资管理的这种难度，但是有的可能确实在应该做的话代价很大，当然跟我们过后的有很大关系，有的东西可能我比如说追求的就像刚才去他可能追求的其实也不是说单个靠运行的效率，他推追求的就是一个很大的scale的这种运算，他把所有的他放在整个芯片上， sm可以很大，所有的model所有东西都放在芯片上，放不下就算了，他就反正能放下的，他就认为是适合他的这种运算。

当然这也是我们所谓古曼思维CC的一个特征，也许某种硬件架构就是个很窄的东西，我们就把它限可能就只能限定在宽带的范围，但是我确实也没什么答案了。

我是很想说反正借这个机会，其实之前也组织过很多很多讨论就是我，我是很想知道就是说呃，我们硬件架构和编译器大家一个更合理的分工，或者在某种某种程度上更合理的分工在一个点在什么地方？现在大家有的从通用到专用，现在专用开始到通用对吧？

一直在循环，反正我是希望就是说这个问题反正不一定很快有答案，但是我就聊一聊也是很有意思的，或者很有价值的东西，行。

说话人16 46:54  
我就说这些啊。

说话人8 46:56  
好谢谢唐博，那么其实今天其实上午已经讲了4个错了，然后我其实已经有点 Over，我们可能需要找个时间消化一下对吧？所以我们现在就有请林林刚华老师，然后首义老师。

说话人2 47:17  
还有。

说话人8 47:18  
还有崔老师到台上来，就坐在这三把小椅子上面。

说话人17 47:23  
然后我们请西鹏来到这来。

说话人13 47:27  
来后我们拿这个话筒来跟台下的观众来进行一个互动。

嗯刚才可能是好。

说话人8 47:39  
那么其实在开始之前我可能要先点两个，因为龙开他可能下午早点走一块，所以先简单介绍一下，张开是给我们自己介绍的。

说话人18 47:53  
大家好，我是幸福计算的张开是幸福计算的联合创始人之一，主要负责软件还有芯片架构这个方面的工作，今天也很荣幸来参加这样一个编译的 b的论坛，谢谢大家。

你有。

说话人13 48:09  
什么问题或者。

说话人18 48:11  
其实因为怎么说分析的问题，就像各位专家所说的，基本上都是传统体系结构里面很多的问题，只是现在大家又找到一个新的这种平安集团的管理，又把它翻出来了，这是最大的一个感受，但是怎么样去在做取舍，然后随着时间的推移，在什么样的一种架构会能够在这种市场竞争中去全国下来，就显得会更有生命力，然后能够构建起一个这种软件的这种生态，这是我们我觉得可以大家会更值得探讨的一个地方。

因为其实从我们跟实际的一些客户去呃交流的这种反馈来看，很多时候我感觉我们的硬件架构设计其实是走入了一个走入了一个误区，就是我去拼命的拼这种极致的信任，对吧？

但是其实在实际的这种应用中没有什么意义，没有真的没有什么意义。相反的话我觉得从如何从芯片这种架构角度去对让软件变得更更友好，然后更易用，我觉得才会让你的产品在实际的客户中会更有生命力。从商业上可能成功的这种可能性会更大，我看到的一些实际的这种问题啊是给供大家各位参考。好，谢谢大家。

说话人8 49:52  
开始看我们对还有一点点点名的要，这个是税源科技的首席架构师刘燕先生。

说话人19 50:00  
你好，这次我们报名晚了，然后我是来自顺源科技的刘院，然后我是负责芯片的架构设计，然后那边的话顾伟说的他是从我们的软件层架构师，然后看看跟大家比较熟悉一点，这边的话其实今天来讲讲我说两句这边的话，其实我因为没有开了准备，然后也是我就带几个问题过来，其实今天我觉得今天各个前面几位像林老师还有林老师发言非常有启发，然后这边的话，其实我们在平时工作时，在提架构的设计的时候，做新的设计的时候，其实蛮头痛的，或者说一个核心的问题，我们会放就说是我们怎么样对存储优化。

其实现在换的话到就是说这个系统看到说。

说话人13 50:56  
现在其实本身算力的一个提升是非常快速的，本身从基金管发展的过程比较延伸，这块是非常快速的，但是最后的存储存入强度问题。

说话人19 51:08  
以及存储的一个话题，最后到就像我们现在看到的特别大模型，刚才也提到了大模型的问题要去解决，这样他的话最终其实会回头到我们怎么样去费用这些费用的一个计算引擎，怎么样去高效的提升就是说数据可以复用性以及可编程就是说已结束怎么样可控，其实大家不太喜欢可控数据这件事，因为这是传统我们CPU来自于CPU编译器的，就体系架构大家都是 h好刚才提到的我记得可能港口唐博士刚才 Facebook那边，我记得他最对体系架构认为对软件编程里面最优的其实有一条cash，然后我们不希望管理，就是说可能很多的 program不希望去管理我们的难度，然后希望能够通过就是说他们的统一变更方式来改来就是说使得我们只关注我们的运算和运算的负载，我以及希望能够自动的通过或者是编辑器或者说硬件上的一些就是说硬件体系结构的一些加速支持，能够把这些东西全部隐藏掉，这样提供一个对我们用户更有优势的一个结构。

其实这块的话也是我们现在看到的一个问题，然后也想把这个问题带回带到会场，我看看大家各位专家有没有什么好的解决方案。

说话人5 52:30  
好。

说话人11 52:31  
谢谢各位专家。有没有？

说话人4 52:36  
这里面的话，我觉得任何复杂系统的设计肯定都来有一个非常关键的词叫催要，我们得舍弃一些东西，才能够获得一些更有价值的这样的一个收益。在商汤的话，其实我见过很多很复杂，从小到说他们写蒜子，大到说现在占领导界的几十亿的。

说话人7 53:03  
每一个过程都要去做这种需要。

说话人4 53:08  
那么对于 AI来说的话，其实有两个点，第一个点就是AI的算法演化非常快，我们回头去看两年，前 AI都是用什么方法，到现在用什么方法之前用的现在的吗？

当然我也没办法，这次可以说两年之后会不会有很不一样的新的算计会出来，所以的话现在有很大的一个fundamental kgun，不在于说AI算法演进的速度以及体系结构，也我叫一个心态非常难，那么他的演进速度和周期完成在一个数量级上，这个时候的话，如果一个芯片它到了最后就这样能够投入到实际应用，在经过了可能一年的时间写算子去优化，就三年之后的时候，那三年之后究竟会有什么样的算式来代表我们主流的模型啊，我们实际上是没有办法去很好判断的。

这里面的话其实很重要的建议就是说芯片也好或者说系统也好，它要有一个相对简单一点的这样的一个设计，这样使得我们的软件站它能够更容易去跟上它。事实上像我刚才所说的，计算所占的时间的整个AI开发流程里面也就是2~3层，然后这2~3层里面你扣10%的这个方法是变成了真正的主要的，并不是有那么大。

真正的话我们希望看到就是说一个新的芯片出来，它能够迅速的形成生产力，这个过程中的话一个相对简化，比如说我这个过程中我简化了它的类型，想讲简化它的整个安全，我可能损失了10%的信用。但是的话我整个软件站的开发，我因此缩短了50%的时间，我能够在更早能够让芯片发挥它的生产力的话，我觉得这个可能是更为重要的一个事情。

其实我并不特别建议说被陈强做的特别的4级5级6级，也许能炸再炸出5%的饱满出来，但是的话我们做框架很多同事非常痛苦，到最后的话呢等他做完了，这个算法就变了，这我觉得那个要是很重要的我的关系，所以我紧接着一个观点，你要说你是不是。

说话人8 55:31  
反对像m二这样打榜的这种行为。

说话人4 55:37  
我是看到了这种在ml上面的很大的，但我觉得这里面是这样子的，我们说说实话商汤也是靠他们起家，我们最早的话打了，因为是2014年的时候说那个包是不够多少，第一个大家关注这所有打法中的模式，最后都没有一个在生产环境用起来，那个是用来纯粹的最大的意义就是说有PR拿投资，我们会这么一个坦率的说班主任来说这个意义主要在这里面，但是它也带来很大的一个事情就是说，在这个过程中，我们其实真正有了一个东西去锻造我们的基础的这样的事情，实际上它是有一个但就是说打榜本身的成果未必是最重要的，在这个过程中呢带动什么事情，比如说这个基础上，我们整个软件在哪个层面，它能够有一些革新，有一些创新，因为一般来说打板的话这个模型做的特别大，对各方面要求的问题特别有挑战，这个过程中的话它会带动一些其他的一些东西进来，这里面其实也是它的价值。

说话人12 56:52  
我觉得其实刚才几位老师也都讲了。

说话人13 56:54  
存储优化确实是一个。

说话人12 56:56  
但现在特别重要的一个英雄，然后现在其实我们之前就针对好几款AI芯片，也都做了相应的这种 memory的上面的这种优化，然后当然我们也会把它就是抽象一下，这个东西其实我们来重新看的话，可能好多年前同样比如说随机分配对吧？

其实差不多的思想，然后但是当然你不能直接搬过来用，但是可以extend一把，然后当然之前大家也做过像 s就是这种什么spm的这种管理什么的，都用这种卡乐什么的是可以搞定的。

但是问题在于不同的架构会有不同的这种设计的，比如说sm的组织方式，然后它的大小，然后它有一些bank上面的约束，所以我们其实觉得有很好的一种思路应该是，当然每个然后每个芯片有自己的工具链，你不太可能说我在像以前弄一个rvm大家都用你的白粉的对吧？

每个公司有自己的白粉的。

所以我们现在在想一个新的photo，就是在做 organization的service，就是比如说你需要做memory了，然后你可以把你的架构输入给一个模块一个service，然后这个service就把分配以后的结果给你，然后这样你再去弄一下，反正因为关于memory优化确实是一个特别特别有意思的事情，包括其实呃呃像刚才唐博讲到的就是这种data flow的定制上面，然后这种routine的定制什么的，现在其实好多人都在做，然后我们也觉得这个东西应该是一个抽象的，而不应该是因为它还是有共同很多共同的技术在里面，就不太能够当时lm其实把很多东西抽象出来了，包括传统的心情看看的是每个芯片有一个，但是大部分的算法其实是公用的，但是现在我们的AI里面基本上就是每人美每家一个，我们觉得这个东西尤其是像memory，这一块其实是比较容易抽象出来的，它的层次化的架构，然后它到底互相之间的互联是什么样子的，这个东西还是有机会抽象出来做一个比较通用的 service的这种模块或者模块也好或者service也好，反正是这样一些东西还是可以抽象的，所以我觉得可能有这样一个考虑，可能也是个好的事情。

说话人5 59:17  
好，所以我觉得。

说话人8 59:21  
 observation service畅想怎么样。

说话人5 59:24  
教学管理方面。

说话人8 59:28  
我们现场还有没有其他的问题。我们今天要准备剧本。

说话人5 59:34  
所以大家非常欢迎。

说话人20 59:37  
先自我介绍汇报一下，我叫王敏杰，我是在亚马逊上海人工智能研究院，我做如果知道的话，我做一个系统叫DJ就把flight专门做出神经网络的框架。

然后我因为现在做的软件上，

说话人7 59:53  
所以就有很多的一些打交道，然后去看说他们到他们在。

说话人20 59:58  
开发过程当中遇到一些问题，比如说我经常发现的就是说一个集合大的问题，因为可以想做认证网络现在还非常的前期对吧？

所以很多的情况下，我发现我们要掉到大的图上，我们发现不知道是算法的问题，还是是硬件的问题或者是系统的问题，也就是说究竟现在不能SQL是用算法制约了还是系统制约，这些我不是很清楚对吧？我就是在思考说这个问题该怎么解决，这是我的第一个问题。

第二个问题是说可能更加的抗生素一点就是说我们是不是需要一个新的language。就是说现在目前来看，我们知道我们现在的长主流的一个变更框架，就是说 Python或者说一个比较长比较好用的 program，language加上test这样一个抽象作为一个编程语言，right，这个模式是不是一个正道路，我们未来是不是应该有新的方向能够比如说把一些硬件的一些opportunity seeks游戏，这是我们两个。

说话人5 01:00:54  
问题很大。

好的。

说话人8 01:00:57  
谢谢。台上没有解答，没有台下也可以有没有接的。

说话人5 01:01:04  
另外的话你。

说话人19 01:01:08  
这个问题的意思是说你希望知道问题在哪，对吧？

说话人20 01:01:13  
解决方案是什么？比如说我现在比如说一个research对吧？我其实在遇到一个比较新的问题的时候，对吧？我有的时候感觉到系统有制约，有的时候感觉到算法有制，然后在这种情况下，对吧？

我其实应该往哪个方向去走，因为我可以比如说出力量说我去做系统，我把这个系统做的比较很大，然后去传销很好，所以说我可能觉得是算法有问题对吧？

可能这个有点偏一点跟偏编译相比，因为我觉得编译可能相关的一些事，如果编译做的很好的话，就可以帮助你说我一个很好的idea我就很快的实现出来了，但是目前我至少没看到嗯嗯行明白了。

说话人9 01:01:51  
我不知道就是说我们一些相当的工具，或者不管是从power value的角度，不管是编译能给你一些case的角度是不是能解决，但是你的问题也确实我我们可以讨论一下。

然后你第二个说法我其实是有点疑问的，就是刚才我在讲的那个东西，现在是硬件可以给你暴露很多东西，有的有的硬件直接所有包都给你，就data plus全包给你，你慢慢怎么用CTRL怎么用，你自己选这边，但这个东西所有的编程的人都愿意这么用吗？这不是给编程人增加太大的难度吗？

说话人21 01:02:36  
包括。

说话人9 01:02:37  
做优化做Khoo的，我不知道你刚才说想想。

说话人21 01:02:44  
我。

说话人9 01:02:46  
也发现有些可能想控制硬件对吧？

有些上层或者说做这个算法，或者说做编程的，人其实挺想控制硬件的，但从我们的角度来讲，做硬件我们还是希望给上层提供的好用一点的抽象，对吧？

说话人21 01:03:05  
否则的话。

说话人9 01:03:08  
反正我不知道这个到时候怎么去解决，或者说你想控制是因为说相声给你的东西不好用，还是。

说话人6 01:03:17  
还是一个什么情况对吧？还是大家碰到一个什么地方比较好对吧？我给你什么东西你最想控制，其他的我尽量帮你去抽象调对吧？

我觉得这样子效率高一点，我不知道这个边界在哪，其实我们之前也讨论很久，就是说。

说话人9 01:03:34  
底层给上层到底包括什么对吧？包括哪个层次。

说话人6 01:03:40  
我的感觉我举两个例子，就是我们来来回回应一下。

说话人13 01:03:45  
就是说。

说话人20 01:03:47  
一个例子就是说我们在呃做算法的时候，有的时候会去开发一些新的蒜子，这些蒜子其实也不能叫蒜子，我可以叫做Micro program，比如说我should transform a rite，我想要把chance for me交给田结构把它给换对吧？

而天都换掉，我其实希望他做很高效，它不是一个普通的论断，比如说举个例子就是Graff做，就是说它的算它为什么是个算字，是因为我在的力度非常细，我要表达的运算逻辑非常细，我希望运算逻辑直接lower到CTRL上去，我在program level我没有办法表达这件事情，现在至少给我的这些program来我都是非常聪明的。

Rite我没有看到说现在有来给我解决问题，我看到一个例子，第二个说我还想提到一个例子是什么？就说大家知道Alex Write， Alex request其实是个非常好的hacker，当时我在过实习的时候有跟他共事了一段时间，然后当时人在用tasteful，然后他说你用素吗？他说我不用说我用qw.com that他很自豪地说我用自己编的东西，我说你为什么要自己编东西？他说我可以去everything，我可以写自己的东西，我可以非常好的去happy到我也好像做了新的东西出来。

Right，我现在看到越来越多的这种，比如说在现在拍照这种框架里面拍到其实做的不错，它能够暴露出可定中华算算不错，但是能不能更进一步，或者说应该怎样提出怎样的抽象给你，development来让我们能够很方便的去实现我们想要的这些idea。

行。

说话人7 01:05:17  
我大概明白了，就是说其实主要的时候需求不一样。

说话人20 01:05:20  
你的需求或者你的创新点可能性没有意义。

说话人4 01:05:24  
所以你希望控制。

说话人16 01:05:26  
直接控制性，就是你对别人给你提供的信用可能还是不太满意，或者说一些新的东西。

说话人8 01:05:33  
对吧？

因为他那种所以其实你想所以你的其实刚才的作为主要的观点就是跟林老师说的，作为或者是在研究搞研究develop和搞生产develop，它其实对于硬件暴露特性的事情，其实是观点不一样的对吧？

所以我觉得我们也开始对吧？

说话人4 01:05:55  
他的合作开始推动还是很快，那么我们内部论战我们就放在下午好不好？

我们大论战我们这个方向。

说话人20 01:06:05  
到了。

说话人22 01:06:06  
发表一下，我就补充。

说话人18 01:06:09  
一下我们在这一块的一些现在正在尝试的做法，我们发现这样一个很复杂的这种神经网络编辑，他对从产品设计的角度来讲，它其实是存在不同方面的需求的，最重要的就是应用性跟极致的这种可编程性跟优化力度之间的这样一个怎么样去做一个平衡，能够取得一个很好的平衡。

对于一般的用户来讲，其实我是很希望说能够非常易用对吧？

最好整个网络编译的过程全部无感，我几乎不需要去介入，但是这个时候我可以接受，比如说 Performance的一些比较一般的这种沟通水平，但是当你度过了这样一个这种新手期之后，对吧？

然后你进入到高级的用户阶段，我希望说对于我这样一个模型能够在你这样一个给定的这种硬件上，我却能够把性能调到最好，这个时候其实又希望说你的神经网络编译器能够去开很多口子，把很多硬件的细节给暴露出来，能够让高级的用户去把控很多东西，从而把婆婆们是能够调到最好。

这是我们看到的两类不同的这种需求，那么怎么样去在神经网络编辑设计过程中，去很好的满足这两方面看似矛盾这个需求，其实是一个比较确切的事情，

说话人8 01:07:41  
可能看来逐渐有一些共识。

说话人22 01:07:44  
跟呃那我先介绍一下，对我是微软亚洲研究院的薛继龙，然后很高兴来我是负责也是作为编译的，然后包括专门项目也是我们做的，然后我先回答一下敏捷的一个问题，简短的回答，然后我再问一下嘉宾老师一个问题，比如说事实上我觉得敏捷是观察到了一个非常有意思的需求，并且我们也是观察到这个需求是一个很大的空间，就是说市场编辑上面在通用性和性能上面，现在的这两个方面是有一个这样的。

然后我们比如说举个简单例子，就是像以酷的为代表，这种算子库它是非常高效的，但是它的灵活性会受到一些制约，向 TM这种compared的它是比较灵活，你基本上上面写什么代码，它下面都能翻译到最下面科了，但是它的编辑时间比较长，那么我们现在市场正在做一个工作，也是看到了这个观点，然后在中间会做一个比较好的渠道，那么我现在不细说了，如果下午有时间我可以稍微简短的分享一下，如果有机会的话，那么对然后接下来我想问一下三位老师一个问题，就是说我们现在也观察到另外一个需求，一个PAD就是我们认为神经网络里面会有一个很强的spac的现象或者一个Python对吧？

然后我们不知道就是 py思考了很久，会不会成为一个非常common的PAD，就说因为人脑的结构这样的，那么首先我想问一下就是说林老师就说在你们在做工业的实践上面，对就是说SARS上面的应用或者落地有没有一些挑战，或者你认为以后有机会有多大？

然后同时想问一下尹老师关于在硬件上面，如果想支持这样的牌子的话，您觉得有哪些挑战或者是不是一个比较将来AI芯片里面一个比较重要的一个技术模块，然后最后就是说问一下崔老师关于在编译优化上面，您觉得有哪些机会或者是挑战，对。谢谢。

说话人4 01:09:43  
首先回应算法方面的问题，就是 spa city这个事儿，这个by nature的话，其实所有的模型包括 c的模型在LP可能动 sparsity pattern是普遍存在的，但是对 pattern的一个充分利用，是不是过高优先级的事情，我觉得也是另外一个道理。

就是说我们现有的其实看上去的话，我非常赞同就刚才李老师的一个意见就是说，其实现在很多的这种AI的应用，它其实 performance是返存这边来办的，他其实在计算上面充分利用18世纪就带来收益有多大，上目前来说还没有特别的清楚。

我们内部其实也有在这方面的探索，就是说可能后面可以去尝试一下，有把18世纪来为全国做得更深度一些，但这个事情其实在我们内部优先级并不高。

因为我们看到这里面它的一个诉求并不是特别的强烈，就是说这里面是普遍存在一个pattern，但它利用这个是不是一个最痛点的问题，我觉得可能还没有到那么痛的一个程度对。

然后刚才其实又回应一下李杰那边那个问题，这其实上是涉及到一个他提到一个就是说你开放一个接口，能不能做很深的special的优化，在牌照这样做啊我们直接的把直接写到汇编这个层面，然后把上级给拿出来去用，实际上它是中间的是非常capable，这个事情是可以做的，要刚才你提到了一个就是说要不要支持他语言，所以语言这个事情的话，其实我在回到香港进入商标之前，其实在MIT毕业之后有一段时间可能在做一部分同学知道他们做了一个语言叫朱丽，这里面的话我常常。

说话人13 01:11:46  
参与创建的工业到期的机会存在的生态啊。

说话人4 01:11:52  
这个语言的话其实从那我们毕业那会开始做了，但是到现在也是这样，七8年的这么一个时间，到现在的话还是一个比较小众的。

就是说这里面就是说我们找一个真正懂编辑的同学和一个小的题目做一套语言出来并不难，这些东西都写在教科书上，这个事情并不难，难的是说你这个语言究竟人家为什么要用，这也是Python绝对不是一个最优秀的语言，很多东西表达不出来，比如说我很多这种类型的信息对吧？

这种编译的语言，比如说处理他能够把每一个类什么类型的什么形状的船都写到镜台上面去了，这里编译的时候就很好用了，你知道它大小数据类型啥都知道，编辑的时候就可以这样做了，然后但是这像Python的话题啥都没有，其实上中间会做很多这种团队的包装，把这信息给提取出来双号，然后去指导编译，但拍摄为什么还是大家去用，我刚才说的就是这个原理跟这个牌套是为什么最后我们在学习刚才是一样的，它特别容易用，而且编程语言有个很重要的一个点，为什么大家用它一定是interactive的，我们写成是以为是说大家写了一大段，几千行200行的编辑，然后就出来一个东西不是这样写程序的研究员，写两行就调一下，然后就不对了，又改一改，它是这样来做作为社区的，所以这个时候的话这种脚本是有语言，其实它是会是更受欢迎，这里面其实在反映了整个领域的催要，就在于说其实应用是在这里面看占了一个非常重要的位置，这也是我刚才一直在我把我抛开的分享里面一直去强调这个事情，如果说语言你要投去做，有两点我觉得需要什么？

就是第一个你真正要投入进去的是要多长的，时间多大的资源，然后它的价值能不能just应该什么，以前我自己在学校里面不会有从这个角度想事情，但是天天现在跟这个公司去要说做报价，都会被这种事情你们考虑就你们投入进去这个东西你做多长时间，然后最后你的名特就是成了我们的家庭化。

我觉得做这个也是一样的，如果是纯粹是说我作为一个最很有兴趣想做这个事，我说没关系，晚上下班了就晚上3:00 2:00把它写出来，我觉得非常好，然后发给他 open source的考虑问题，但是如果是作为一个formal的investment在公司里面的话，我相信的话你总是要跑到上面给钱的那个人，你为什么要去投入做这样的一个事情？

所以我觉得更大的是这个点，如果是graph，因为level这个事的话，其实我是觉得说最终还是得有一个落地的地方，落地的这个地方。

第一现有的工具其实比，你落地的 desire的 step是不是有比较大的量比较低的点？第二的话就是落地最终产生的是影响的，受众的范围是不是足够大，这两个肠道一起去，那可能就是说你毕竟你最终的目标有多大，所以判断的问题电商。

说话人2 01:15:10  
从芯片角度看，我觉得可以毫不夸张的讲，就是说过去5年里面所有学术界做的出来的 AI芯片，能够有比较好的application，其实都是在搞surpris。

因为坦白讲除了src以外，在计算角度来讲，其实没有太多新东西可以做。

当然主要是我们从底层看，它的贡献还是很大的，因为咱们 spa这里其实有两种，一种是为什么model里边。

说话人16 01:15:46  
不是你什么产生的，还有一种。

说话人2 01:15:49  
 FV这里边，因为你有录入这些东西，它自然就产生了很多了，而且it这里边的 SQL这里它还是动态，你也不知道他在哪。

所以这时候从做 research的角度来讲，这就是很好的一个事，对吧？他很demo，然后他又量又很大，然后你可以设计各种办法，在硬件层次上，就是你软件上你编译上你去完全不知道的，因为它是WiFi，所以你可以在硬件上设计各种的电路结构或者各种方法，你就看一看然后你，比如说甚至你都可以改一些底层的乘法器什么这些，让他能够在遇到这么大的SAR anthrax的时候，他能够效率更高。

说话人6 01:16:36  
所以其实过去我们看到我们里面很多。

说话人2 01:16:40  
非常紧急的工作，其实大家就都是在搞这个，如果要是统计一下的话，性能提升还是挺明显的，从底层计算角度来讲去对。

说话人4 01:16:50  
芯片提升还是挺明显的。

说话人2 01:16:53  
但反过来讲就是说实现 Spa的检测它代价挺大，因为实际上就是你在一个摊子里边你要去检测随机的0，然后你检测了以后还不能是说影响这个。

说话人18 01:17:15  
计算的冲突。

说话人2 01:17:17  
因为你检测本身它其实是有个复杂的逻辑在那里，然后这个逻辑如果处理不好的话，他实际上就把整个延时，你check本身的延时比计算的延时还高了，所以那就也不值得了。

这样最后的结果有可能检测部分的面积会很大，在形变面积会很大。

所以其实从商业的产品看，好像目前看 a100里面它是在model上面。

说话人17 01:17:47  
它现在支持自己，但是他 spa Citi还是一个regular。

说话人2 01:17:51  
训练的时候你去自己搞，然后你完整的甚至包括 APP这里边的sport，好像大家现在也都没怎么考虑，但从研究的角度看，这个东西我觉得还是很值得做的，因为从性能提升来讲，效果还是会很明显的，但是就去产品角度来讲，去去平衡一下面积功耗，包括指数的各方面的总的来讲，我觉得还是很值得在底层的处理。

好。

说话人12 01:18:23  
我觉得这边一方面psat也是一个非常热的问题吗？我刚才想了一下就几位老师两位老师在说的时候，我稍微想一下，可能我觉得直接研究的有三个地方，一个可能本身就对I'd这种model，它可能有一些sparse的这种特殊的格式，因为以前大家TS二个他这种格式更多的是从以前比如说图或者是p这些数据来的，对于a来说它的数据特征可能有一些自己特殊的地方，像我们之前有一个博士生做过一个工作，就是你像这是很典型的，你图片过来他可能有些维度是系数的，但是比如说它是不系数的，反正这种可能就有一些机会，你可以去设计一些新的数据格式来处理它，我觉得这是一个机会。

然后另外一个机会就是往硬件上VIP的时候，那一个就是说由于现在一个是像尹老师本人提到的，现在本身就有很多针对spa这种硬件，当然有一些预测你怎么办？把 MAPI上去它也会有一定的格式的这种问题在里面。

然后另外一种我们现在做的更多的可能还是说把sparse的应用MAPI到像GPU这种，它更多的是为单词来设计这种上面，可能你这种时候怎么去把这种 shared memory用好，然后把尤其现在像vector vector现在对技术有一定的支持以后，这种的像今年MIT好像有一个paper，IV跟 paper它反正也是有一些这种方面，它可能当然不是特别的全，但是有一些然后那嗯另外一些我们现在在考虑的，其实s PAR s ity precision结合的就是你这个硬件上本身它有好多有不同的因斯特因特巴的哈佛的，然后不同精度的这种数据类型，我们在spa的时候是不是可以跟数据类型交到一块去做这个问题，我觉得本身这个东西蛮有意思的。

说话人3 01:20:20  
好谢谢。

说话人8 01:20:20  
我要提补充一下，我们组也做了不少司法工作，这些事业研究尹老师说的学术界都喜欢做这个是吧？也有很多领会的文章，我们现在也是逐渐去考虑往编译器，作为一个主要原因什么？就是说我们反正这种paper大家还是基于仿真对吧？像不管是Michael paper还是大家可能等等，那么仿真其实都是不准的，哪怕你做再严谨参数调再严谨的其实还是不准的，因为还是软件方面，所以这是为什么我们开始去往编译器这个地方在编译器去考虑叙述的问题，然后我一旦开始做，刚才特老师提到像c32这个问题，我们首先要开口的，或者后面怎么去把它圈成一个，因为底层是这样子对吧？

你肯定 Transit肯定得在前面。给它做好，所以我们就发现一旦开始做这个事情，工作量就急就急剧的上升。你就是说你基于TM好，还是基于上班错也好，我们可能pm就要做一年多时间才能够有一个实验结果，所以我觉得我还是比较看好这个方向，还有其他问题吗？

说话人22 01:21:36  
好，大家好，我是徐欣然，是现在负责煤改电开发的这一块的，然后我这块有两个问题，一个问题是说我们最近做的比较多的国内的各种芯片训练芯片的对接工作，然后这块其实我们就会有疑惑，就是说呃目前对接界面一个比较合适的点在哪里？

像金子鹏金老师这边 msp这边都是今天我相信应该是一个全方面的对接对吧？

算字库语音甚至各种l应该都是足够开放，但是假设是比如说一个比较通用的芯片，我们对接是应该在可以出现一样彻底，我把一个图交给你我不管了，还是说像sra这种稍微细一点点的，还是说语言级别，还是说提供一个算词库，这个感觉层面其实很多，但是这件事我们对接的一直很痛苦，就是不管是sra还是说三字库，可能这个问题都会比较多，因为毕竟使用者可能特别喜欢，还是说然后在Python里开一个 PPP，然后我随时都可能打个断点把 co m改一改接着往后跑，对这种事儿其实是一个非常困难的，所以这是想问大家的看法，然后尤其是在语言这一块，比如说汉武帝的放心，我相信应该没有太多人写过，真的真的写过忘c这么一点，因为可能所有去。

说话人13 01:23:00  
尝试写这件事，包括。

说话人22 01:23:02  
我这边设计的要不要太复杂这件事，可能国产不会不会有太多人有意愿去学对吧？

我学了一个非常复杂的精巧的芯片设计语言，最后我下面工作好像用不上，可能很多人就会说我不想去学这个生态，就会是这种情况。这是第一个问题。

第二个问题是说我们在实际业务中可能安徽嘉宾老师这块的一个点，像一些长尾算词，还有一些日常用的传统CV算法，实际在模型中用的还是很多的，比如说像我们公司一些检测模型，往往可能百分之四五十的时间是花在这些地方，真正的标准的网络结构部分，其实这部分大家优化的都很好了。

而前后那几部分其实一方面是说工作很少，因为他不是很有意思，就没什么可做的，另外一方面是我们尝试了各种现在的编译器，其实在这方面都没什么太多的投入。

我相信比如说现在我有一些复杂的CV三组操作怎么feel起来一个事我们试了TM还在的，还有太极都，还有我跟你说我想知道大家怎么看这件事，是不是觉得似乎现在搞深度学习编译器的同学们在躲避这部分的工作，这是我的一个疑问啊。

说话人8 01:24:25  
谢谢您其实第二个问题主要原因是什么？因为你这些长尾算子二八定律，长尾算子搞半天，其实最后数据结果不会不是那么漂亮，你也大不了拍一本对不对？所以肯定没有人愿意看。

说话人4 01:24:41  
首先我刚才在学这个图，这点跟张老师说的不太一样，我们观察到常委算战略，相当的在真实的c模型里面，肯定如果能够解决这个问题的话，发行费我觉得应该也是非常有机会的，但是要解决好，就是说刚才那个图我们有个同事是我们就在餐馆里面做编辑，就是说为什么说我们像他可以建那么多做系统，做计算库的，做编辑的，做框架做这种。

说话人13 01:25:11  
这样的东西好。

说话人4 01:25:15  
胖是应该也是类似这种情况，关于就是说回到刚才说的适配国产芯片的事情，其实商家内部这也是我们一个非常重要的事物，我有可能过了11点的话还得去另外一个厂，就是一堆国产芯片厂商在里面。

这里面的话很痛苦的一点就是说，我们对接任何一款芯片。

说话人6 01:25:40  
都得从头做不一样的事情，是吧？

说话人4 01:25:44  
不是说这还谈不上，是有一套标准的设备跟界面，完全不存在每个芯片的怎么做法都不一样，然后更重要的就是说当时深有同感，两个月变一次API，这个事情会非常聪明，代表你几个月干的事情或者在来弄，所以在大概。

说话人13 01:26:06  
记得这个工作调整一下。

说话人6 01:26:08  
是一个非常重要的。

说话人7 01:26:11  
所以的话从。

说话人13 01:26:12  
我们的角度来。

说话人4 01:26:13  
就是配5款芯片。

说话人13 01:26:17  
首先判断。

说话人17 01:26:18  
还是首先。

说话人13 01:26:20  
是这个公司要不要跟我合作，我认为真正成功的思维不是说我们定一个方向。

说话人4 01:26:28  
interface在这个地方我交给你，后面我不管，我输了这个东西绝对做不成，如果真正要把这个东西是配好做出竞争力的话，肯定这两个团队需要真正有这样一个共同的call跟go就是说把这个东西做出来做好。

然后的话大家要建立这种选配置的这种信任，真正我还是去才能把这个事情做好。当你过ST的时候的话，你的肺在哪？我觉得这个其实并不需要一个非常value的事情，就是在摸索的过程中就会形成一个机遇的题材，形成了一个face，这个其实是更重要的，所以基本上的话这里我们有很多的这种国内做芯片的厂商也找到我们，然后我们首先第一个要判断就是说你是不是真正有诚意去合作，然后基于什么样一个场景去打磨这个事儿，最终我们去查的时候，真正把这个事情给做成了。

说实话，实际上我们去跟芯片厂商谈这个过程叫找对象，但真正谈好了要适配了那就结婚之后的事情，我觉得老师。

说话人13 01:27:39  
有新的问题。

说话人17 01:27:40  
我先自我介绍一下，我是张先生。

在商汤负责训练框架，对，然后我IQ一下这边的第二个问题，其实我这边在公司做各种模型支持的时候，也有完全这个就是类似的现象。

拿大家都能够看得到的例子来说，现在我们开源了，我们的检测框架才能离开现场，现在影响应该是相当大了，然后这个仓库其实已经优化过很多轮了，即便是在这样一个优化的很好的检测处理，我们在做训练的时候，他的时间有30%是花在除了看看我们和别人都以外的地方，都是那些所谓的后厨里面小蒜子30%。

说话人23 01:28:28  
这还是一个优化很好。

说话人17 01:28:30  
事实上有更多的就算把研究员在自己想自己的idea的时候，设计出来的这些写的这些自定义代码，它的时间会更长，可能百分之四十几是吧？

这是一个客观存在的现象。

然后我们还有一个观察，现在的芯片厂商他们都花大量的力气去堆特色库，然后 Tensor的算计就是一代比一代都恨不得多一个人出来。

然后刚才的前面几位嘉宾的分享，关于架构或者编译的架构分享，我看到硬件和编译的注意力也都在 tensor或者是这种com的计算上，包括像前 NV发布的新一代a100，它比v100的新可能翻了一两番，但我们在实际网络上它能表现出来的性能提升其实只有不到20%，就成网的提升不到20%，这个现象其实是我们看到非常痛的地方。

然后也有一些工作，比如说体检他们能做这个代码生成，然后也能看起来好像能把这些肠胃的酸子能够生成好去做，而我们也有去我们我们有是tbm包括我们自己也有开发类似 tbm的扣件的工作，但实际的表现这些工作到现在都还不能够被算法研究员直接很好地使用。

就是说我们现在还不存在这样一个工具，能够让算法研究员自己就把30%的 cost降到降到他实际算力所应该占有的品质，或者说稍微大一点其实很难，然后我觉得是需要讲出来的在做其实有很多做编译的做架构的专家，我觉得大家要重视这个问题，这个里面接下来很可能我们的训练所要有进一步的性能提升，其实不是在 pencil或者提升上，是要解决后面肠胃怎么去把它给优化掉。

对，然后我还有一个点，我们其实做过关于小算子编译的工作，在一些有限的case上，我们可以做到说我们的一套工具就是类似特别的那样的 API可以把一个代码用户写的原来写的Python的那个标签能够加速10倍或者20倍，这个形式是有可能的。

其实我觉得如果大家把注意力放到这边来的话，应该是能够做出很多很好成果的。

说话人6 01:31:08  
我看总的道理的，刚才他几十位的都没有，还是一个数据之外的，对，现在艾特我的，所以说我们回顾一下就是说。

说话人17 01:31:17  
那个框架跟。

说话人13 01:31:20  
今天的边界的问题。

说话人3 01:31:22  
所以我谈谈我的理解，因为跟一个产业的发展是有关，就是说任何一个新的产业，它肯定会经过一个首先然后到水平分成，就是说我觉得这样的产业还处于一个垂直整合的阶段，它其实还很难分出来。可能过了若干年以后跟整个产业成熟了以后，它的结构才能慢慢的把它稳定下来是吧？目前看的话其实就两类接口，一类算是个接口是吧？这是原来不但给我们的接口。

第二个可能我们大量的dsa芯片出现以后是吧？因为我每个dsa芯片它肯定要它的目标肯定是要超越 GPU的性能是吧？所以他不得不多大量的这种但是括号包括大量的图层的封装，但是我觉得未来的话我觉得这两层接口应该都会在，所以说另外一个维度，我也觉得这个是框架一个新的机会，因为大家知道最早的时候就它的flow它是静态图，然后后来白涛就搞了一个动态，但是未来比如说我们大量的dsa的芯片的架构出现以后，其实纯的静态图其实在性能上又发挥不出dsa架构的它的一些效能是吧？

这个也是为啥我们还是包下去做这个原因之一是吧？因为你你芯片用不起来，你得自己搞一套架框架去把它发挥出来，我觉得未来可能整个的框架的价格它是非常灵活的，既能够说我们企业底下的像这种库拉的这种算式的接口，也能去对接这种npo的接口啊，这是第一个这个问题。

然后第二个是刚才现在提到说那种所谓的长尾的算术，其实跟前面我觉得敏捷提的问题有点相似，就是说这种其实我们整个编译AI编辑面临一个问题，就是图灵完备的问题，就图灵我们怎么做一个图灵完备的切我的性能还是可以的，这样的一个编辑，实际上我们都没有做到，因为整个崖边及Tudor zui科技就是一些kennel，然后上面我们的应用一些model，我怎么样有一个基础的原子的一些科目，然后我的model、通过我们、编译器我都能自动生成。

如果我们做到这一点，我觉得整个算法工程师的效率就高了，他如果需要一个新的算子，他利用这种原子的组合一下就行了，但确实这个挑战非常大，我觉得下午我们可以好好讨论一下。

说话人6 01:34:04  
我们下午分组讨论。

说话人7 01:34:09  
有升高密度的控告，然后关于算法架构这块，还有什么其他的？

大家好，我是那中科院软件所票CT实验室的赵宏斌，然后我平时是做MR这块的工作，然后大家刚才提到比如说dsa或者dsa我觉得ml可以帮助大家把中间打开，可能还要要做，然后我这边的一个问题，因为我们做端到端的一些编辑的任务来看。需要很多工具链来协同配合的，是不是应该要很长要有翻译的过程，先看logo的过程，然后再翻译到lvi之类的这种过程。当我们在第8个一个一条呃路的时候，我们可能很难定位到具体是哪条工具链出了问题，比如说我合理的结果不是很理想，所以我我应该从哪里开始给bug，我平时比较头疼的一个问题，然后也想请教一下各位老师有没有好的方法？

放一下，首先来看郭老师一下，对，有时候特点结果或者不是很理想或者分类的不对，然后你说精度就是他跑上这个东西，不知道具体是比如说是训练的参数问题，还是具体编译的过程，就是哪个算词编译的过程出了问题还是就很难定位了，就在我这个事件当中。

说话人4 01:35:47  
那个是形成两个回应，因为知道我在三趟的话，大量的时间就放在所谓换了一套东西，换了一个风暴，换了一个什么东西，又差两个月。

说话人17 01:36:01  
又要调几个月，当然我们形成了一些方法，让我觉得这个事情

我说我这个问题其实问的严总呢，但是确实很广泛存在，我这边讲一下我这边的经验，我这边主要是做训练，所以主要面临的问题是说，如果硬件或者是编译链某一个地方发生了变化之后，整个训练的结果发生了发生了偏差，而且是最难的，其实发生了微小的偏差，这种情况该怎么处理？

对我们现在的话，首先训练这边其实已经它的计算过程已经是两个定义的，它的训练是由算词拼成网络，然后网络在做 s GD或者是这种往前前向的迭代，然后再进行多轮迭代之后，得到的，结果每一步是其实分阶段分得分得非常清晰的，我们在做问题定位和定位的时候，基本上就是顺着这个层次从底向上先看它的每一个算子是不是行为跟我们对标的一个能复现的结果。

比如说在MV上是不是一样，接着再看它的整网就是成像出来是不是一样的，再看它的梯度是不是一样，再然后去对它的一轮迭代之后是不是一样，这个过程中还得干掉一切的随机性，然后进行这样的比较之后，就可以一层一层的去发现他的问题，直到说到最后整个能够选下来进度如果都能够对齐应该就能齐了。

然后这个过程的话，我们是在实践中形成了一套就是工具体系，可以去在每一个层次去看它的计算的结果是不是能够跟前面在一起来做到快速的去这个问题。

说话人20 01:38:00  
能发多少人群。

说话人7 01:38:05  
这是第一个问题。

说话人17 01:38:06  
这个倒没有说我们专门花多少人去做这个事情，我们从开始做框架，就开始多多少少会遇到业务这样的问题，然后我们会去我们其实一直都在解这样的问题，然后之前解最开始解决的很痛苦，就是第一个刚来的这个问题可能花了一个多月，才能够找到一个很简单的原因，后来的话我们逐渐的就把这里面的经验和这个过程把它逐渐的工具化，到现在一有一套工具出了问题上来，我们按流程一步一步服务的，按工具来，基本上大多数问题都可以做得很好，对。

说话人7 01:38:43  
我就问一个问题，我刚看到你的这些验证方式都是数字化的，有没有比如说编译过程中形式化的方式出现？

没有，但是大同也是这个问题，对，一个是说。

说话人22 01:39:05  
我们没有做形式化，但是我们会去做比如说优化之后做很多的这方面的测试，一个优化完了之后的等价性的测试会起来很多，对，然后另外一方面是不知道问的主要是训练还是推理，推理我们这边推理的可能是比较多，其实跟刚才张先生说的差不多，其实就是这块的工具。

说话人7 01:39:28  
很重要的就是整个优化，不能说我做优化公司就结束，优化工具可能需要很多其他的工具去支撑它，比如说你的优化是不是能逐级展开，而不是一次就只有0和1，对吧？

你可以一部分逐级展开，这样的话通过一系列的工具可以帮助你去乘以什么去定位这个问题，对，然后其实你不知道这个问题比较多，包括这些东西最后怎么才能。

说话人22 01:39:55  
从一个优化后的图再能应用社会原始的图，这些都是需要工具去支撑的，拿着一个完全优化后的和优化前的说对不上分，那确实没什么办法好。

说话人17 01:40:09  
啊嗯不是对，我是阿里巴巴派团队的，我叫刁南松。

说话人7 01:40:15  
这个问题可能是在座。

说话人13 01:40:18  
各位的不不专职三位嘉宾。

说话人7 01:40:22  
是这样，刚才提到就是说。

说话人17 01:40:25  
今天我们做编译优化以后，它的正确性还在不在？其实一方面就是说我们通过述职方法，其实在工程上是比较现实可行，就是说大部分问题确实能够查出来。

但是嗯如果要100%保证正确性的话，可能还是确实需要一些比如说形式化的这样一个工具，比如说这个符号计算这样的一个系统，这样的一些系统以后，能够保证说实实在在还是对的，就像硬件设计里边形式化验证工具一样，我们现在有没有公司那么，目前或者说高校有没有在研究这方面的这个话题？

对。

说话人6 01:41:08  
我们做了。

说话人12 01:41:14  
点点，但是对我们现在也是比较苦恼的，也是想拿出来跟大家一块讨论，这个是其实当时我们做寒武纪的时候遇到的一个问题，就是它的芯片只有低精度的，但是它要真的它支持的不光是DNN的model。

说话人6 01:41:28  
它还有一些其他的生产线model，然后这样就会遇到，比如说我最简单我在规约。

说话人12 01:41:33  
的时候就会出问题，但进度就不够了，然后我们就做了一堆这种补偿什么这种，但是后来又遇到a他又做另外一个算法的时候又不够了，然后又得补偿，后来我们就做了一个工具，就是说什么样的你原来原来拿到的肯定是个float的这种程序，然后我们做了说你转成转成比如说哈佛或者英特8以后，哪些哪一个区域的代码会导致你精度会影响比较大，这是一个问题，但是这个东西我们做了框架，也是一来一多次跑啊跑，然后最后定位到一个东西，其实有点像虽然说是自动的定位bug定位，但是问题是他要让好多次要在斯明瑞这里面转好多次，也是个特别恶心的事。

然后我们就去也是想用sim包来可以玩游戏来做，但是后来去做的时候就问了，就这个东西比较难，是因为保利k一来了，医生本身他不这是浮点的什么浮点进来他就搞不定，然后所以这样的话这确实是一个特别open的问题，我们现在也刚开始做，然后其实不是刚开始京东时间我们做了一年了，但是现在我们也没想好从怎么来克的角度能把这个东西分类它好，上次也跟软件所另外不是敏捷，是另外一个老师，他们也讨论过这个问题就是说本身关于floating points的在包的问题就是一个现在就没解的一个问题，所以我们也不知道这个解在哪，我不知道别的团队还有没有在做的。

说话人8 01:42:59  
你觉得你来他说不是啊啊。

说话人16 01:43:03  
不是除了这个模块的其他部分的性质原则，其实我问题就是ml它其实。

说话人9 01:43:12  
 m二它其实最后会转换成很多的 Dyk或者层次的 Lorraine，然后他每次都认为反正他也会提供一些验证方法，但我我好像没看到形式严重的方法，对，所以我不知道就是说我们以前做芯片做ada对吧？

形式验证这个是特别重要的准备，你基本不太可能通过仿真或者像软件的这种测试方法来测，我感觉以前变编译这块应该是形式验证也是一个我们做eda的时候是形式验证，是继承这种编译的这种软件的一些东西来的，我不知道这块这块是现在有什么新的问题，还是有一些。

说话人20 01:43:56  
但是像。

说话人12 01:43:57  
其实以前欧洲有个团大家问的比较多的，像。

说话人6 01:44:02  
你编译前后的代码是不是等价的，对吧？

然后应该华为冯青云老师一直在做这个，然后。

说话人12 01:44:10  
当然他国外的邵东老师他们也一直在做，但是这个的问题在于他的他对你输入语言的要求要比较有一定的限制，他不是说你随便写成啥他都能过的，然后对于编译器这种。

说话人6 01:44:23  
里面写的乱七八糟的这个东西它就不太好介绍。

说话人20 01:44:26  
欧洲有一个团队曾经就是自己重写了编辑，然后它编译器它证明过了，但是那个里面有好多优化都没有开。

说话人6 01:44:36  
但如果你要想把 l都不用mil比如说把lvm能挣过去，可能就已经比较难了。

说话人9 01:44:46  
这可以一一一堆的形式验证它的要求是挺严格的，就是你有写法的问题，你还要有很多。

说话人16 01:44:54  
这个点要对上什么什么的确实是这样。

说话人22 01:44:58  
其实这个问题的本身是想说。

说话人3 01:45:01  
假如说我换了硬件以后。

说话人7 01:45:03  
我进度。

说话人13 01:45:04  
不一样了，大家可能都会。

说话人3 01:45:08  
是吧？

大家都有，如果是因为硬件变了，确实那个东西我觉得好像跟形势解决不了这个问题，因为它这个是一个对比标杆的问题，因为现在举个例子，你硬件你这个进度对不对？

并不是说你本身对或者不对，而是因为你跟 GPU去比是不对的，但其实对于深度学习来，你本来就是概率的东西是吧？没有说对正确或者。

说话人7 01:45:34  
错误的说法。

说话人3 01:45:35  
或者比如说你跟传统的比如说HPC的你做了一个软件，你发现跟你原来用的软件精度不一样，你说谁谁对呢？这不好说，但是人家已经用了对吧？你必须跟他比如说你最后把你这个软件条条条太一样，你就认为是对了。

所以这一块是我觉得好像不太像是形式化能搞定的问题，但是如果说你编辑做出来，你正确一个或者超系统做出来正确一个，反正我们有很多工作都在做，对。

说话人22 01:46:07  
其实我关心的是什么就是说。

说话人3 01:46:10  
就说我们。

说话人24 01:46:14  
无论是芯片还是说这个编译器，我们有可能会改变计算的顺序，浮点的话，你计算顺序稍微变了，比如说a加b加c与ab先加和BC先家，出来可能就不一样了，然后我们最后看到说结果不一样，我们到底要有个判断说，实际上是我们不仅仅是计算顺序带来的微小的偏差，而是我们实实在在我们编译已经出问题了，像这种东西也是有价值的，我们现在就看到一个数值，我们不好判断说到底问题出在哪，是如果说形成那个符号计算，它能够告诉我说你放心，这仅仅是计算顺序的问题，对吧？

但是如果说没有这样一个东西，我仅仅是数值。

说话人13 01:46:55  
计算一样的，我就很难判断。

说话人16 01:47:00  
埃克兰的问题，这里面有可能一些我是杨军以前的阿里巴巴现在暂时是可能还在毫米7，对，然后可能有一些Tracy的部分腐败死了这么多问题，比如说我们说到-0计算的话，它具备一定随机性，我在GPU里面可能a加b加c我换个新建的话，可能c加b加a这样的话就会不一样的，那么这种情况一般来说是很难去真的去通过比对精度去看的。那么当我们引入了像IP16，h6的话更复杂，我们可以换个角度就是我们的bug的来源在什么地方？

其实bug来一方面可能是金融的误差，另外是逻辑的bug，如果逻辑bug的话，我们所做的事我们控制随机性以后， bug就能够稳定的出现出来的。那么如果我们对我们的data pattern作为一个fix，我们控制好我的输入机构的话，我每举不同的test le，我在多次去验证以后的话，我的误差在稳定的一个点出现，那么可能会去办理。如果在不同点出现的话，可能更多的可能是一个随机误差，但是这个不是很精细的形式化的方法，更像是一个比较工业性比较脆弱的玩法，但这个方式可能在一定程度上专属南通那个问题。

另外一点就是说据我自己了解的话，可能在分为刚开始做特色号的时候的话，他们其实是拿IP公司做一个golden在里面做的很严格的check，也经过了很长的周期的话，来确保新店的记录是符合要求的。也就是可能我这边一些还过来问一些一些问题，好。

说话人8 01:48:26  
我们最后要不由我们就下午继续，我们最后就有一个问题跟给听众，因为我们跟嘉宾下午。

说话人11 01:48:37  
我说一下 NB，我是来自天天树之星做软件库的，然后我回答一下envy是怎么的，MV他的调换顺序a加b加c和AA加c加b包括电话数据，包括他其实有些svo它会调 ethnic，因为原子加它计算顺序其实会它你看，其实有些算法它是分确定性和非确定性的，你设计艾特MAC的时候，它的一些算法的顺序其实就会变，那么比如说我就会演，然后NB验证的时候，它其实你那些ABC的它碰到的问题其实会有一出的问题，其实因为路程加起来。

说话人13 01:49:16  
看起来我当时记错这个问题，所以他 QQ他去见的时候，他里面他提出来的时候只有几位，比如说只有00014这些数。

说话人11 01:49:28  
那么他就能保证他加的时候，他那个位置进度是控制好，你这样的话你可以用这些实验相当于一个类似于形式实验，跟他说你可以有一定程度上也可以。

然后我想提的问题就是说。

说话人1 01:49:41  
我。

说话人11 01:49:42  
我现在做一些可能的优化，比如说我出了结果，比如说我可能是对这个坑的结果是我们其实做的时候是比较考虑到时间上，对这个算是时间上优化比较多，但是其实也有问题，你最后看到联合在一起之后，你跑的可能两个，单位款不一样，就有些客户他运用的但大家看比较多，有的可能用的量比较多，然后有的可能看一看行为，你就感觉意义上有些不太可控，因为他开始可能速度就上来了，开始行为其实不好预测，那么我架构我怎么会是说，而且还有一些还有看到的问题是看到你的输出去的量，你的输出形式其实在很多输出形式它是有百家姓的，比如说你的处理在领导之中，他可能是nch他不也不能nhipoc那么对于我的段子来说，我这两个都可以处理。

这是我一这一部分比较快，因为我下一步就会减半。

下一步另外就算是处理非常可以的，这种的话在上市场的框架是怎么搞的？价位在最后一句话总结一下你讲的比较多，我的主要问题就是说上层框架怎样处理我那个多个坑，比如说我看到写出来之后，其实多可能都有个各自的特点，有的memory访问少有的就这样。

说话人16 01:51:16  
这样子现在我们介绍看一下我的回答。

有一点点。对，其实本质上是两个问题，第一个问题就是说我的不同的算有不同时间版本，我应该靠哪个版本来去做全局更多的优化。第二就是说当我需要跨算我确认的话，我该怎么决定说我的feel是过于激进，还是说可以做的更接近一些？因为可能确实不是越做越多越好，就像李老师说的情况，你说的更多的话可能是有代价的，不是乱吃。

那么第一个其实很早以前就已经有过类似的做法，比如332t里面做的比较a类更接近，那么它是把整个组合优化变成了一个热力代数的问题，变成了一个其实用一个Salo来解决东西，我可能定义成一个类似于一个板栗Gramm的依赖性，每个链的每个节点的话提供一些保持，可能是ncw可能是awc甚至可能加班需要量化的这样的一个量的转换。

那么在整个链的讨论过程中的话，它通过一些组合优化的求解找到最后的路径，这是他的做法。那么TM的话应该我记得去年发了一篇文章。

说话人25 01:52:19  
应该。

说话人16 01:52:19  
是adc是铃木的工作，他们是做了一个咱们头干的做法，求经四季也是类似的情况，其实这个问题本质上如果我们把它抽象一下，它其实很像是ACM里面一个DP的大问题，这个可能是一个问题。

而第二你的前一个问题关于这个费用的话，有些算组对我们的memory有一定的业务很多，那么这里面本质上就是当我配合完以后，我带来收益的同时，我的科目的资源消耗增加了，因为没法利用我的科特的Barry来保证说每个科室独立的可以用所有资源是吧？

那么这里边的话有些不同的方式， Fc里面做的比较满意，一起他做了一个比较简单的规则了，他们说我该不该费用，我是在说考虑收益，那也有些工作做的可能更为漂亮一些，就我们是cosplay的，我决定是我设置完以后的话，我预估的收益会有多少，cos有多少来决定，怎么做一个探索。

但这个问题可能比较吃一点，是说可能在NV的v100的架构里面，它的收益的上届不是那么的高，因为修正本身它的班级配置可能会有绝对位置，这个问题从v100往后以后变得更有圈儿的，因为整个算力的增加跟我的访中浪的概率的一个显著，可以包括像NV的新的软件站，包括现在很多dsa的公司的话，都在考虑类似的问题，希望会更漂亮的一些QQ的工作，大概这样的一些情况。

说话人25 01:53:38  
好。

说话人16 01:53:40  
我们已经超过10分钟了，然后我们上午的环节就简单在这里就结束了，然后我们下午11:30再回来重新开始，然后我们吃饭的地方在嘉宾其实已经发给大家了，然后其他的听众就有了，你们自己解决了。好的，好，谢谢各位。

说话人6 01:54:05  
有来下午我就打车，另外一个群里面好就一块走出去，我有家。好，是现在才到。把分叉的这边问一下。好是吧？相对好一点，应该都是比较觉得应该等会我要问一下，好好。问一下我们小孩没在，对我在发给大家的时间啊。我看是吧？因为比如说你告诉他，你讲一个事情，你说这个问题要这样去外面。18年过去了，对面对好来家国企方面，我今天讲的更多，谁知在网上来说，这就是我们可以通过对吧？就的是好不好？现在牵引来，大家知道这个东西这里面给了好多书或者干什么东西，一个经理之前对客人呢直接说清楚了，去招商的问题，知道他技术问题的。

好的技术我们来说上海不过我们这边。好的好，咱电话的话特别你就想什么新作用，他们写了一些调整的，然后在那一一楼搬起来没有，我也不喜欢去找一下。

除了还是报下面这个问题，你们今天讲，然后很我知道了，他们要做工程，关于第8个问题，就是说都是很实际的，我觉得很好，我们大家可以想一想，对方面工作有没有可能这个是同样的就是这个是上届毕业的学生，然后当时就想先谢谢老大。应该当时你知道这个项目下来了，就相互跟一种意见，而且去专门的方法，你可以先去做起来每天然后做几年反正就稍微拍成同一点硬件加在一起对。

不是，然后等一下还有很多的，而且关键是现在对客观的，我觉得新东西算法本身它是不是10%，他就不应该搞个西方上海这个也不知道对还没到这个团队怎么说下一个对对。我们不了解，

说话人26 01:57:56  
由于你们在这里吃饭吗？看一下在你们知道吗？

说话人6 01:58:11  
好像也不能吃饭。

说话人26 01:58:15  
上面的公司。

说话人6 01:58:20  
我们不用打算。

说话人26 01:58:23  
把产品的。

说话人6 01:58:25  
企业还有问题，我觉得然后要这边吃饭在一起，现在就在25号，在所有的东西是不是一定要先买发票的？二楼可以买。可以的，一块对。自己看对其他人就。