Cassini: Network-Aware Job Scheduling in Machine Learning Clusters

Sudarsanan Rajasekaran and Manya Ghobadi, Massachusetts Institute of Technology;

Aditya Akella, UT Austin

NSDI April 16-18, 2024

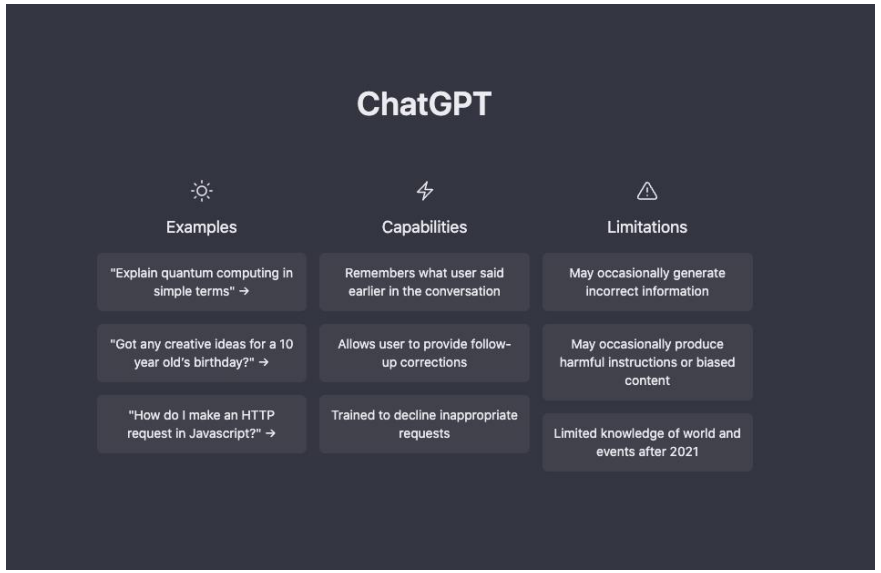
汇报人：孙铂钛

2024年7月18日

Content

- Background
- Related work
- Design
- Evaluation
- Conclusion

Background--Rise of large Deep Neural Networks (DNNs)

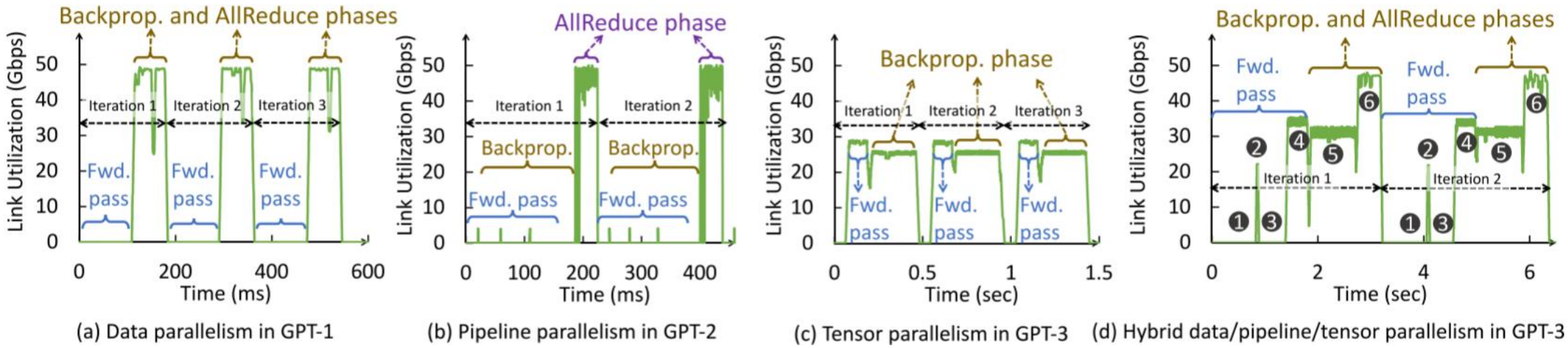


ChatGPT

Sora

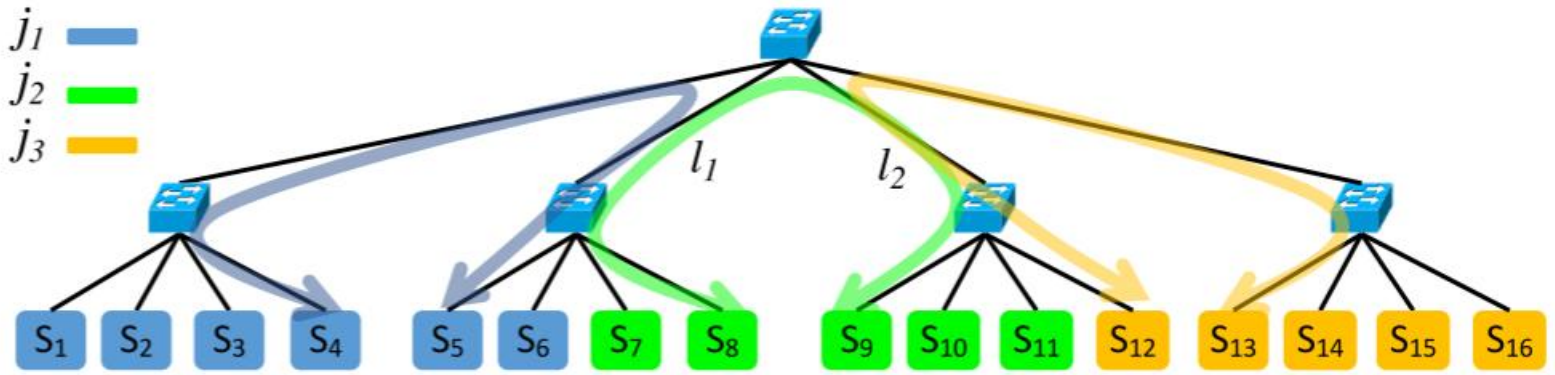
- 随着数据集和模型规模的增长，对高效的GPU集群的需求日益增加。
- 在分布式机器学习训练中，随着GPU数量的增加，通信开销在训练迭代时间中占据了显著的部分，这成为了影响训练效率的关键瓶颈。

Background--The traffic pattern of different parallelization strategies



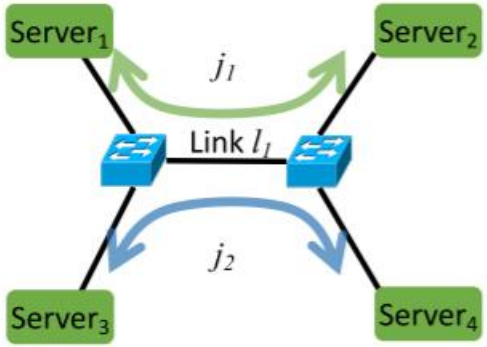
- 尽管不同的DNN模型可能具有不同的通信需求和模式，但许多模型的网络需求在迭代中是重复的，只要训练超参数保持不变。
- 一个迭代的网络需求可能包含多个上升和下降阶段，这些阶段的确切幅度取决于并行化策略和超参数。

Background--Network sharing is inevitable

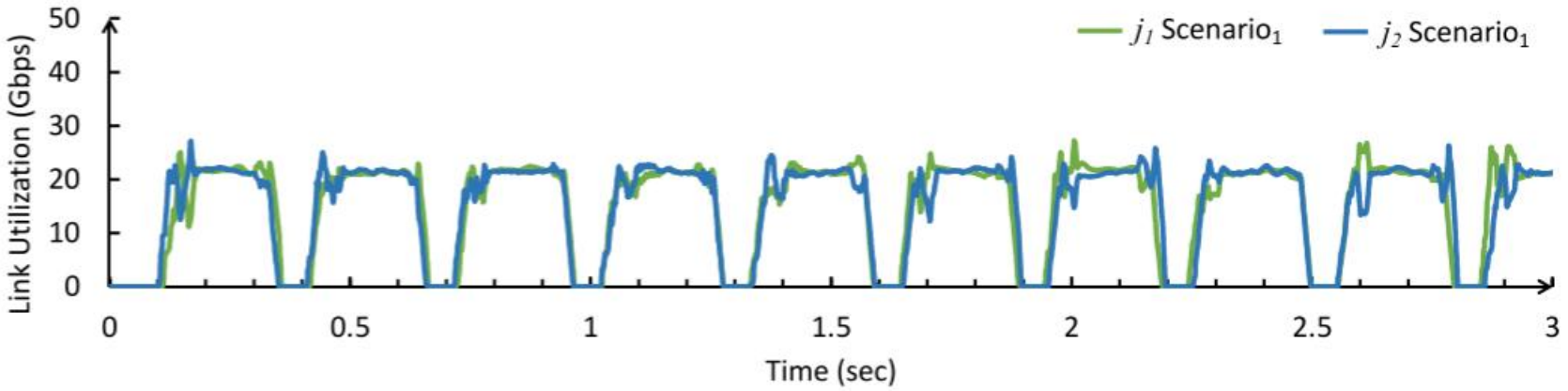


- 忽略通信模式会导致拥塞和网络利用率不足。

Background--Interleaving communication demands

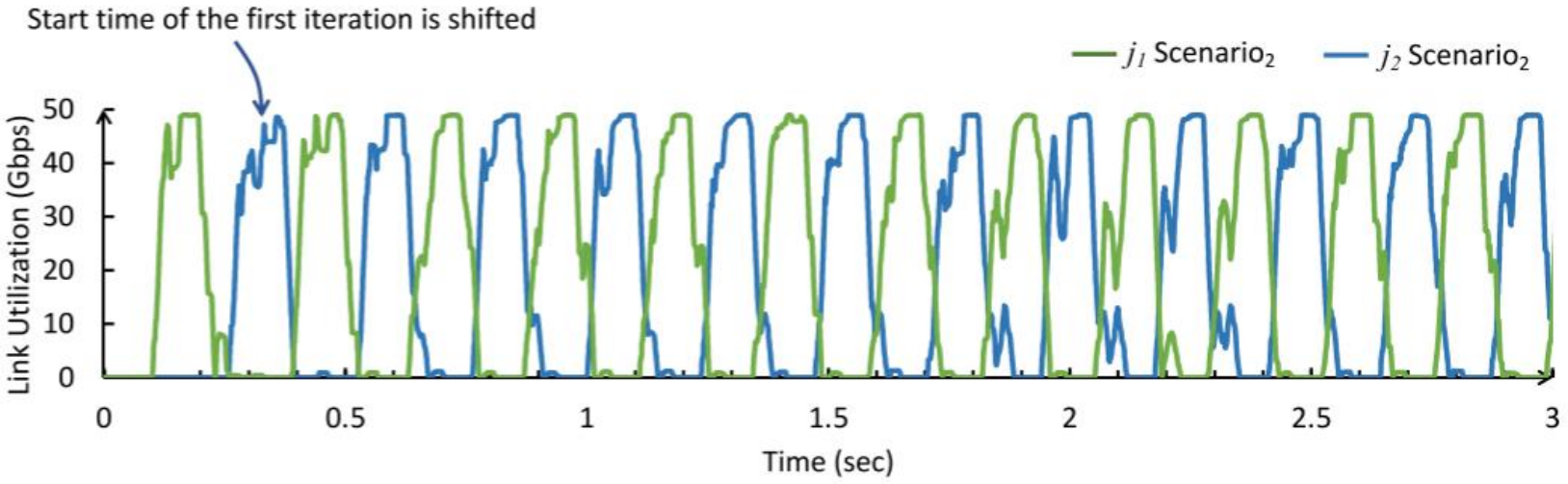


(a) Experiment setup with four A100 GPU servers, 50 Gbps links, and Mellanox RDMA NICs



(b) Scenario$_1$: jobs start at same time

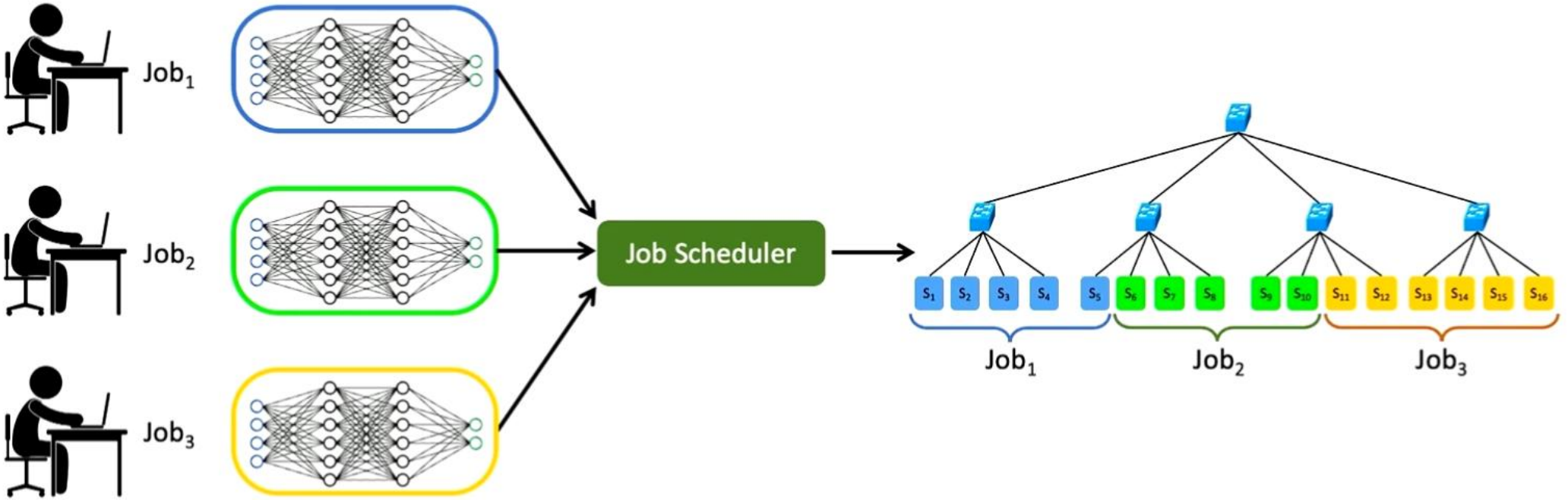
- 交错的通信模式有助于加速 DNN 训练。



(c) Scenario$_2$: the start time of j_2 is shifted by 120 ms

Related work

- 根据拓扑邻近性放置服务器
  - Themis (NSDI2020)、Pollux (OSDI2021)、Gandiva (OSDI2018)
  - 通过将同一作业的服务器放置得尽可能接近来减少网络共享，但它们没有考虑在放置服务器时交错不同训练作业的通信模式，而主要关注计算资源的分配。



Related work

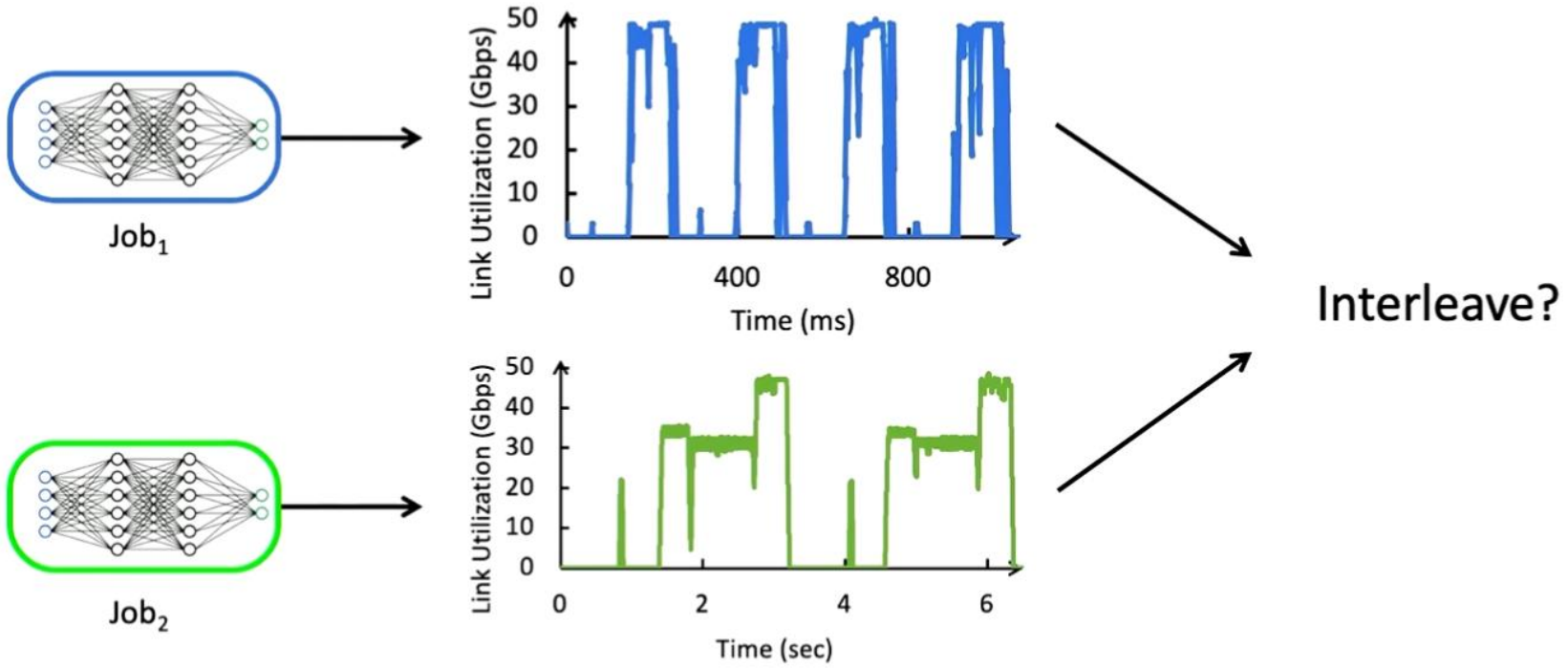
- 多资源共享
  - Muri (SIGCOMM2022)
  - 将DNN训练作业的关键资源(如GPU、CPU、网络、存储)交错使用。然而，Muri的资源交错方法只适用于共享同一组资源的作业，只有当一组作业共享同一组gpu时，Muri才能交错处理一组作业的计算和通信阶段。
- 通信感知调度
  - ByteScheduler(SOSP2019)、Syndicate(NSDI2023)
  - 通过调度和优化不同GPU服务器之间的通信操作来加速机器学习训练，但并未考虑跨训练作业的拥塞和网络共享。

Design--Cassini: An interleaving-aware DNN job scheduler

Challenges:

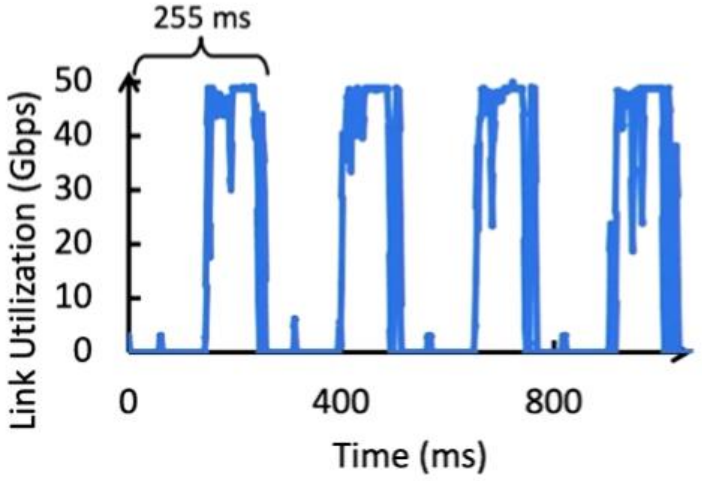
- 如何确定给定的作业组合是否可以进行通信需求的交错?
- 如何实现这种通信需求的交错?
- 如何将这种方法扩展到大型集群中?

Design--Challenge1: How to determine which jobs to interleave?

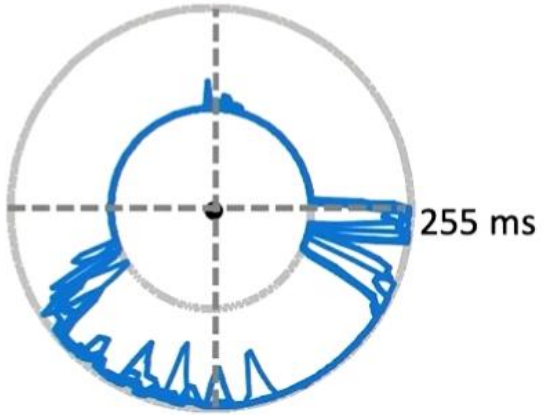


- 需要在多个作业的数千次迭代中检查是否能够交错。
- 不同的作业具有不同的迭代时间和通信持续时间。

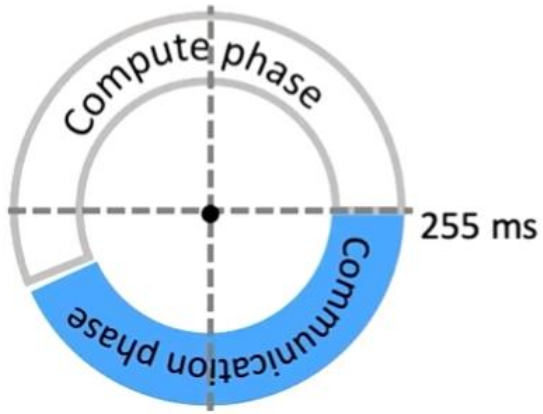
Design--Geometric abstraction



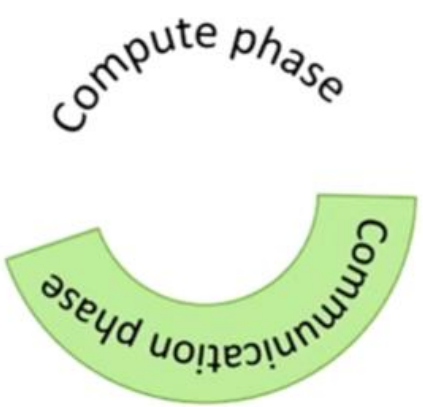
Network demand



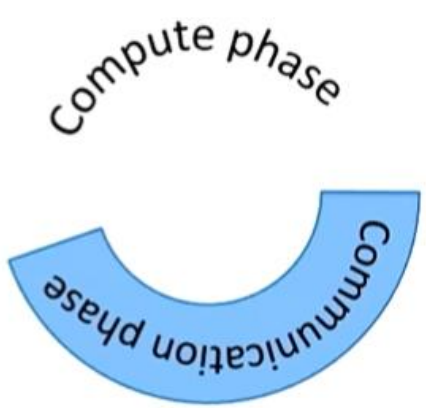
Network demand rolled around a circle



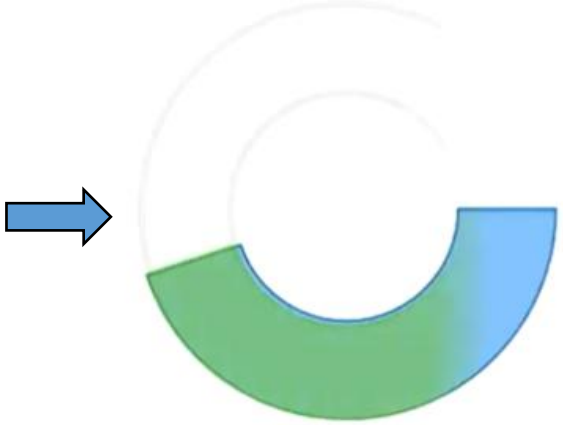
Geometric abstraction



Job_1 : DLRM



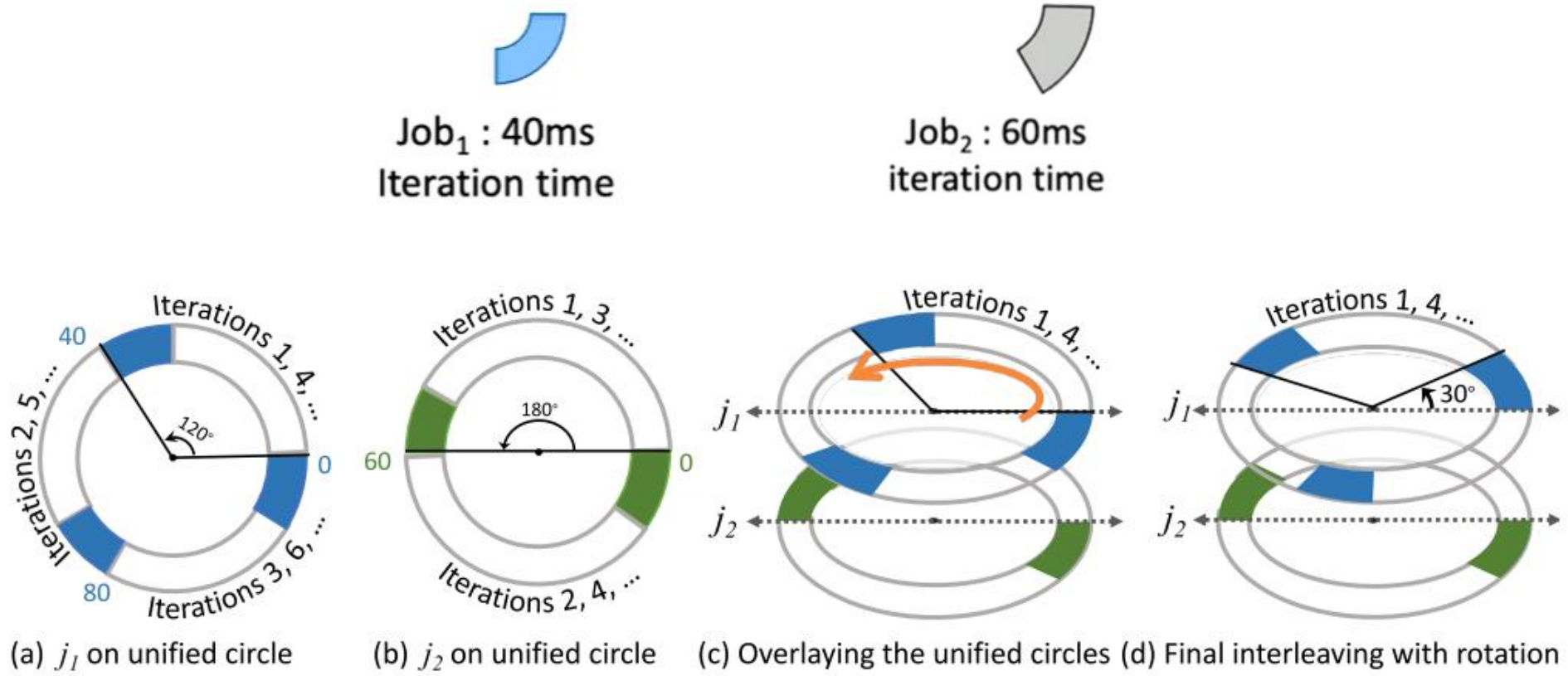
Job_2 : DLRM


Job_1 and Job_2 overlayed

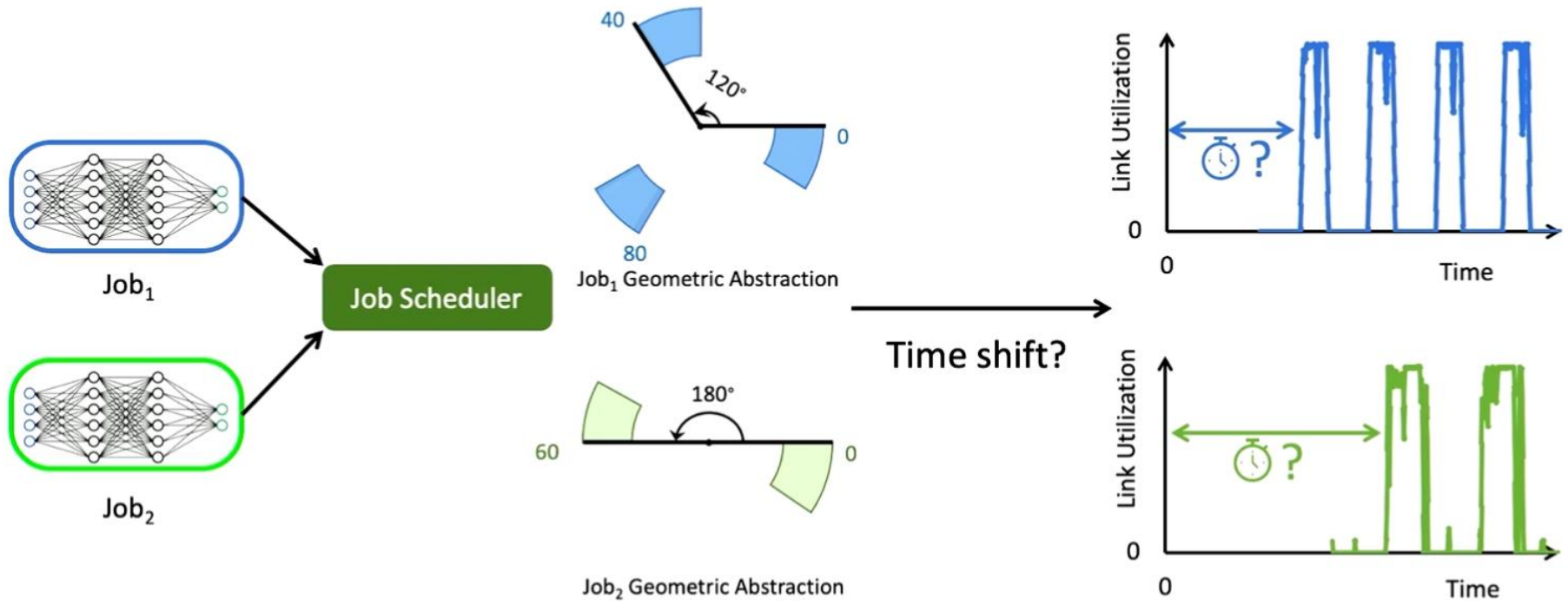

Job_1 and Job_2 overlayed

Design--Jobs with different iteration times

使用迭代时间的最小公倍数来构造统一圆



Design--Challenge2: How to enforce interleaving?



- 每个作业在开始第一次训练迭代时应该移动多少时间?

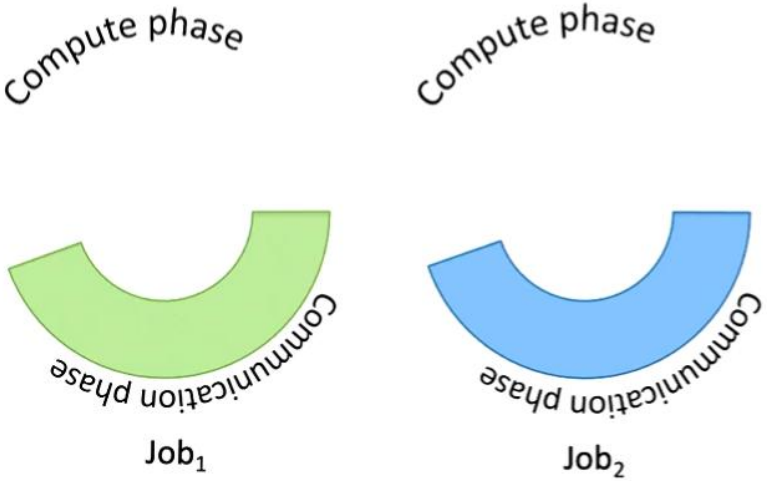
Design--Optimization formulation to compute time-shifts

- 优化目标：将每个作业的统一圆叠加并旋转，使所有角度的多余带宽需求最小化
  (即兼容性分数最大化)
- 输入：同一链接上竞争的一组ML作业
- 输出：每个作业的旋转角度

兼容性分数： $score = 1 - average(Excess(demand_{\alpha}))$

$Excess$是特定角度α下所有作业的多余带宽需求

当多余带宽需求为零时，兼容性得分为1(即100%兼容)。



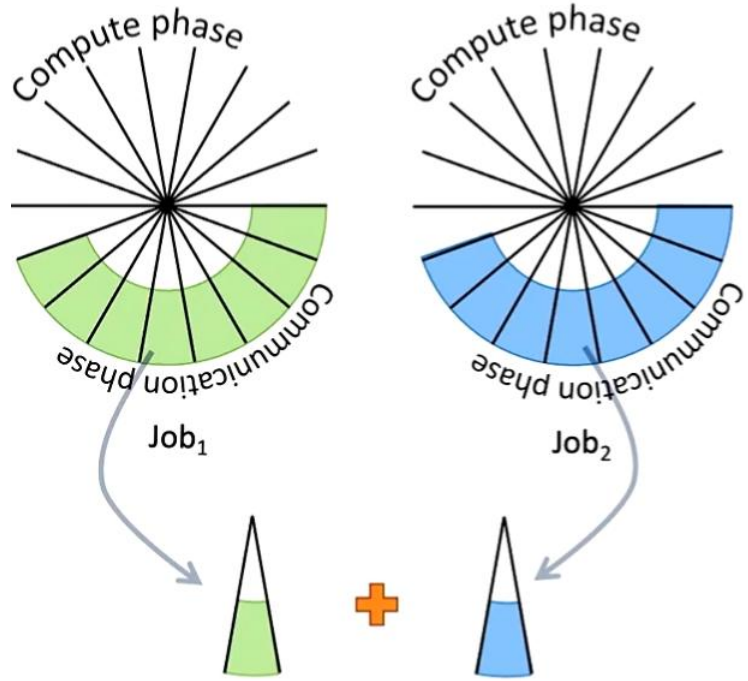
Design--Optimization formulation to compute time-shifts

优化目标：将每个作业的统一圆叠加并旋转，使所有角度的多余带宽需求最小化

输入：同一链接上竞争的一组ML作业

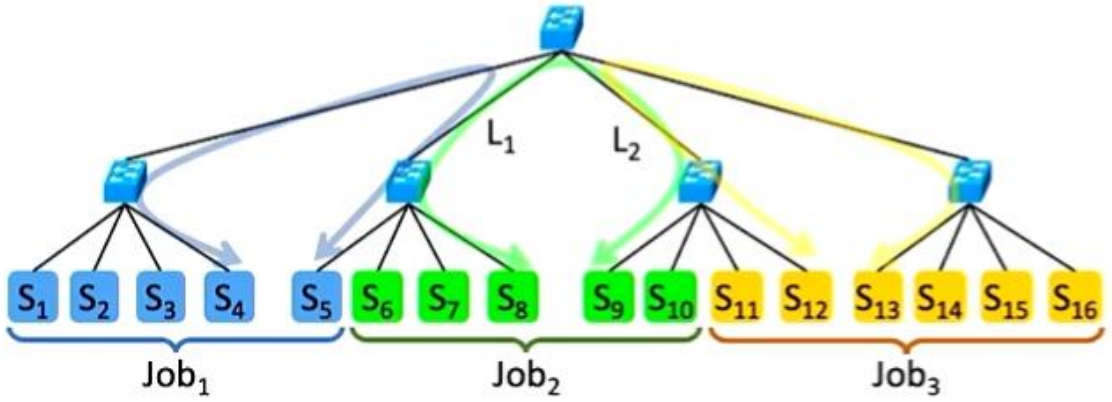
输出：每个作业的旋转角度

- 将圆离散为更小的角度
- 遍历所有角度：计算所有圆的带宽需求总和
- 计算每个角度的多余带宽需求
- 将旋转角度转换为时移



Design--Challenge3: How to scale to a large cluster?

- Job_1与Job_2在L_1上竞争，Job_2与Job_3在L_2上竞争

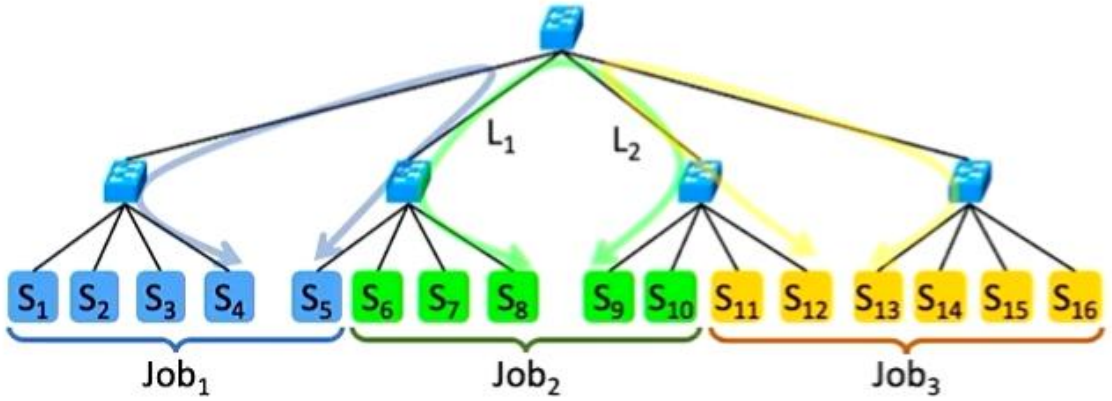


Job	Link L_1	Link L_2
Job_1	200 ms	-
Job_2	300 ms	600 ms
Job_3	-	800 ms

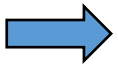
- 问题：Job_2在两个不同的链路上有两个不同的时间偏移，该如何解决？

Design--Relative time shifts

- 确切的时间偏移对于通信交错并不是必要的，我们更关注不同作业之间的相对时间偏移。



Job	Link L_1	Link L_2
Job_1	200 ms	-
Job_2	300 ms	600 ms
Job_3	-	800 ms

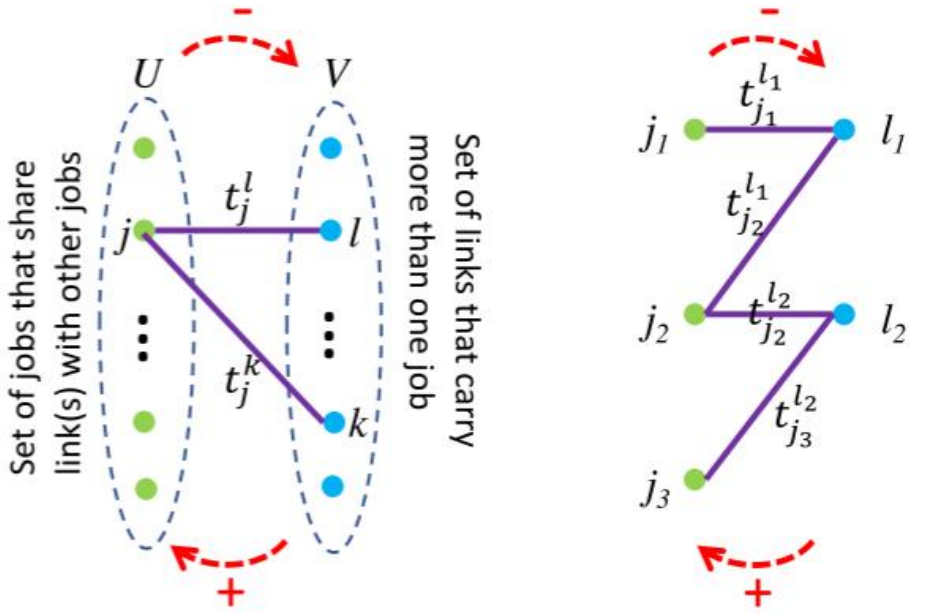


Job	Link L_1	Link L_2
Job_1	0 ms	
Job_2	100 ms	100 ms
Job_3		300 ms

- 使用相对时间偏移来依次为作业分配时移！

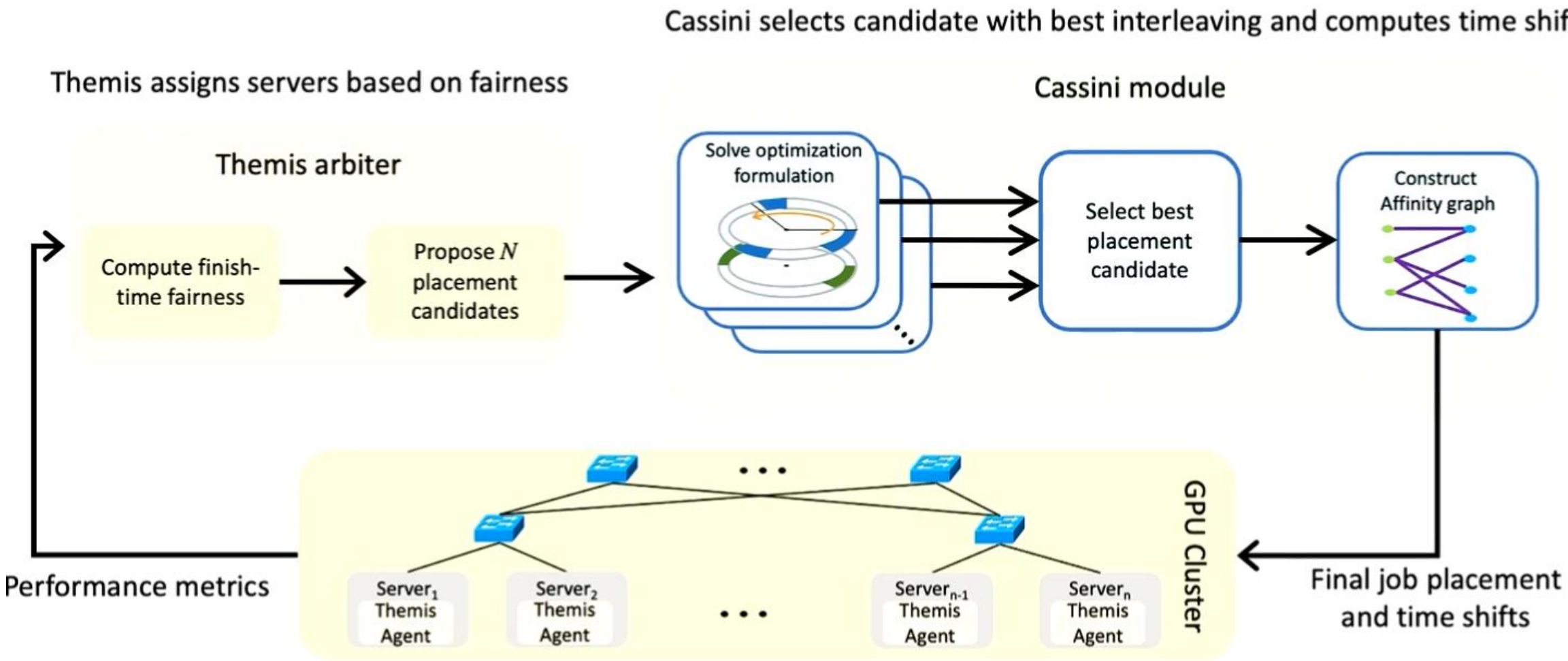
Design--CASSINI Affinity Graph

- CASSINI引入了一个二部亲和图 G = (U,V,E)，其中U和V是两个顶点集，E表示U和V之间的边集。
- CASSINI使用图遍历算法遍历亲和力图，为所有作业j找到唯一的时移t_j，同时保持所有链路的兼容性。



(a) Bipartite Affinity graph　　(b) Affinity graph of Figure 7

Design--Augmenting Themis with Cassini

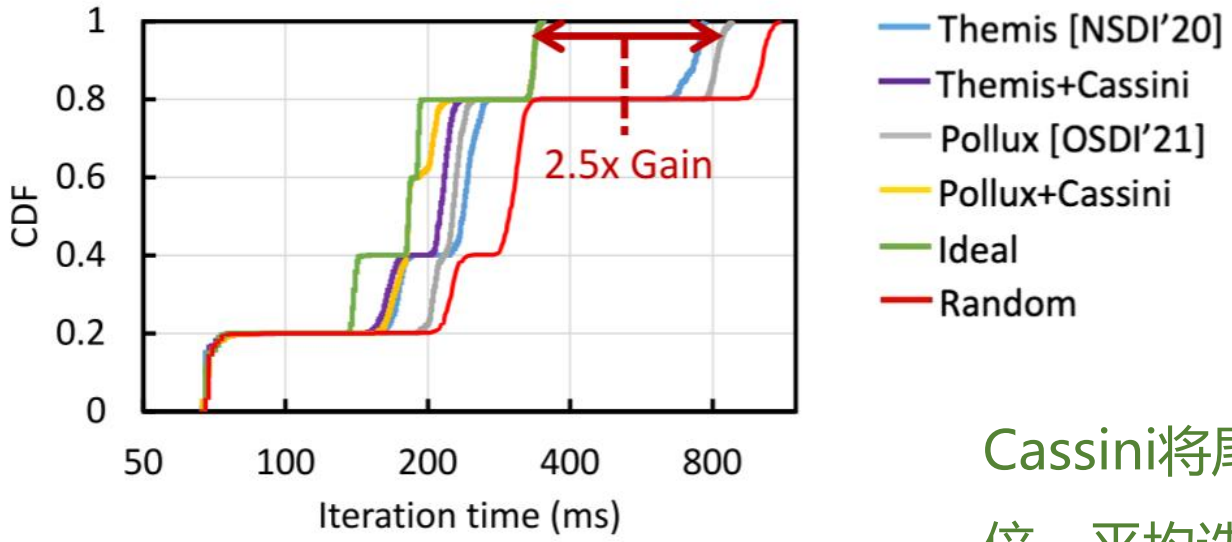


Evaluation

Setup:

- 爪式拓扑链接的A100 GPU服务器集群（24台）
- 50 Gbps 支持RDMA的网络链路
- 13种流行的DNN模型，涉及图像分类、NLP和推荐模型

Evaluation--Cassini speeds up iteration time

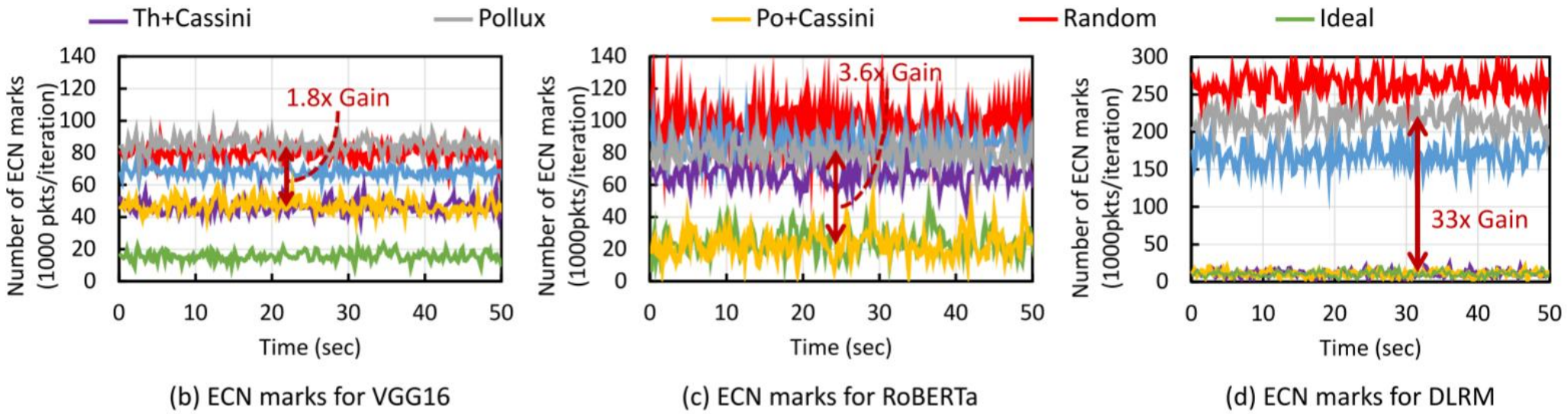


(a) CDF of iteration times

Cassini将尾部迭代时间提高了2.5倍，平均迭代时间提高了2.2倍。

- CDF
  - 表示在横轴指定时间及以下完成的迭代所占的累积比例，范围从0到1。
  - CDF曲线越陡峭，表示迭代时间分布越集中。
  - 可以识别尾部延迟问题，即那些完成时间特别长的迭代。
- 尾部迭代时间
  - 完成时间较长的迭代。

Evaluation--Cassini reduces congestion



- 每当网络出现拥塞时，就会创建带有ECN标记的数据包。

Cassini有效降低了带有ECN标记的数据包数量，说明能够减少网络拥塞的发生。

Conclusion

Cassini：

- 引入了一种新颖的几何抽象方法，通过将时间需求“卷”在一个圆周上，来模拟不同作业的通信模式。
- 使用亲和力图来表示作业和网络链路之间的关系，并通过图遍历算法来为所有作业找到唯一的时间偏移值，同时保持它们在所有链路上的兼容性。
- 定义了一个兼容性得分指标，用于衡量不同作业共享网络链路时的潜在交错程度。这个得分有助于调度器决定如何放置作业以最大化网络资源的利用率。
- CASSINI设计为一个可插拔模块，可以与现有的ML调度器（如Themis和Pollux）集成，增强它们在调度时考虑网络需求的能力。
- CASSINI的设计不需要交换机/网卡的特殊支持，如预留和优先级，也不需要改变拥塞控制协议，使其易于在现有的硬件环境中部署。

Conclusion

Limitations：

- 系统假设每个作业在网络中相对独立，且作业间的通信需求不会相互影响。这可能不适用于作业间存在复杂依赖关系的情况。
- 在动态变化的环境中，如作业频繁到达或离开，CASSINI可能需要更频繁地重新计算时间偏移和调度策略，这可能会增加系统的复杂性和开销。
- 如果作业的通信模式发生变化，或者出现非周期性的通信需求，CASSINI可能无法有效处理这些变化，从而影响调度效果。
- CASSINI在设计时可能针对特定的DNN模型和训练策略进行了优化。对于其他类型的模型或训练策略，CASSINI的效果可能有所不同。

Conclusion

Ideas:

- 扩展CASSINI的调度策略，同时考虑CPU、内存等其他资源的调度，实现更全面的资源优化。
- 考虑其他领域的调度问题，例如在云计算环境中，是否可以借鉴CASSINI的几何抽象和亲和力图的概念来优化虚拟机的网络通信和资源调度。
- 是否可以进行泛化，将CASSINI的调度策略应用于其他类型的分布式系统，如分布式数据库或大规模并行处理系统。

请各位老师和同学批评指正！

2024.7.18