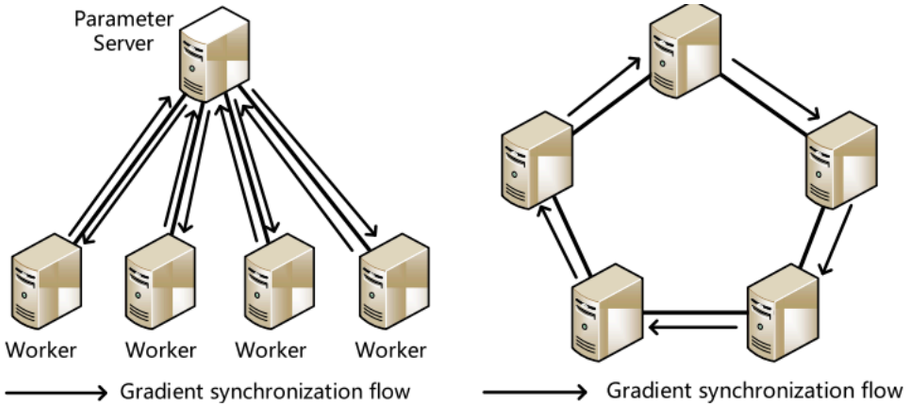
A Scalable, High-Performance, and Fault-Tolerant Network Architecture for Distributed Machine Learning

INTRODUCTION

- GPU数量的增加能够减少训练时间，但是GPU数量过多，导致通信时间的增加大于训练时间的减少
- DML网络的要求：1）可以大规模部署。2）同步的时间开销降低 3）容错机制

MOTIVATION

- Fat-Tree架构的同步开销大，可能会造成网络拥塞影响的扩大
- Ring架构的有单点效应，没有容错机制
- BCubeML可以同时降低同步开销，实现容错机制



(a) PS algorithm (b) Ring AllReduce algorithm

Fig. 1. Gradient synchronization algorithms.

Fig. 2. A Fat-Tree network with 16 servers.

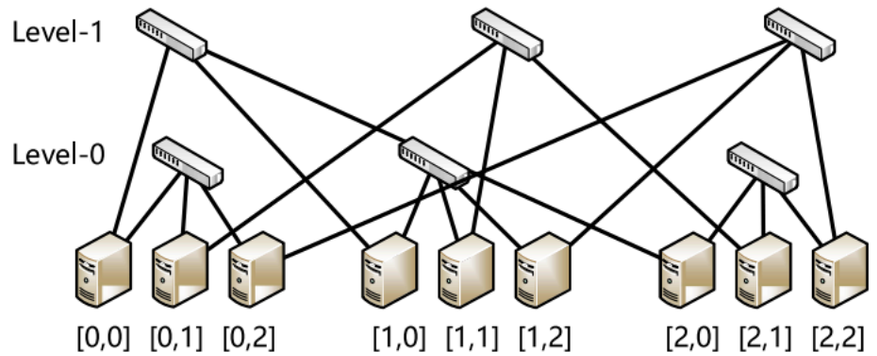


Fig. 3. The topology of BML(3,2).

TABLE II
COMPARISON BETWEEN DML NETWORKS

DML network	Theoretical GST	PFC range	Fault-tolerance
Fat-Tree	$\frac{2*(N-1)}{N} * T_F$	whole network	Yes
Ring	$\frac{N-1}{N} * T_F$	1 server	No
BML(n,k)	$\frac{2*(N-1)}{k*N} * T_F$	$n-1$ servers	Yes

BML Design

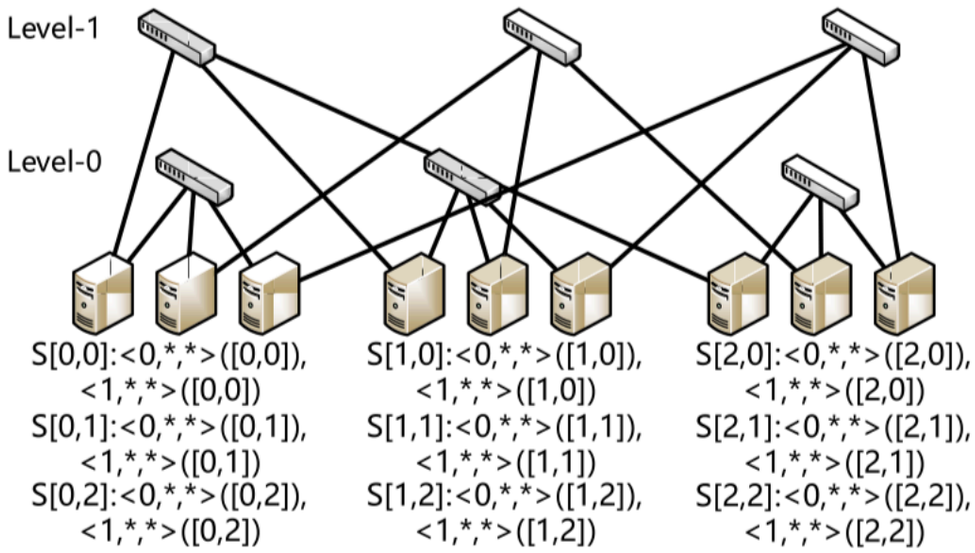
BCube

- $BCube(n,k)$：有n^k台服务器，$k*n^{k-1}$台交换机
- 服务器表示为$[v_{k-1},\ldots,v_1,v_0](v_i \in [0,n-1],\forall i \in [0,k-1])$
- 梯度分成$k*n^k$片，分为k个线程进行聚合。
- 梯度片的理论通信时长$T_C = \frac{T_F}{k*n^k}$，T_F：全梯度的理论通信时长
- 每台服务器上的梯度片表示为
  $< e_i, v_{k-1}, v_k, \ldots, v_0 > (i \in [0,k-1], v_i \in [0,n-1])$

BML Design

BCube

- $BCube(3,2)$：有9台服务器，6台交换机
- 主机表示为：[0,0],[0,1]…[2,2]
- 梯度分成9*2=18份，
- 梯度片的理论通信时长$T_C = \frac{T_F}{18}$
- 每台服务器的梯度表示为：<0,*,*>[0,0]，<1,*,*>[0,0]，代表服务器[0,0]拥有18份梯度信息，
- <0,0,0>[*,*]表示为每台服务器的<0,0,0>均汇聚到服务器[0,0]上

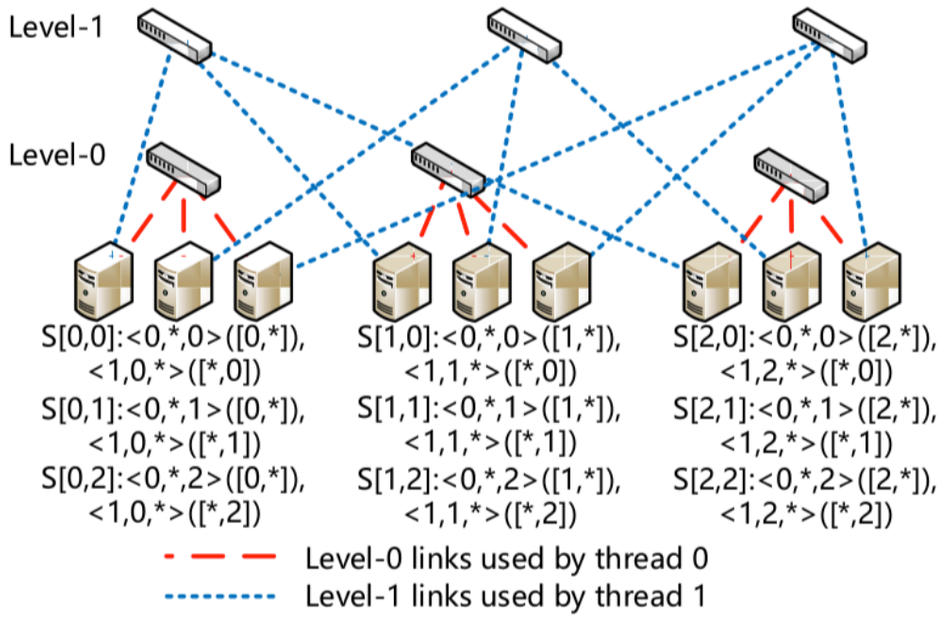


BML Design

Aggregation1

- 线程0:
- [0,0]传输<*,1>[0,0]给[0,1]、<*,2>[0,0]给[0,2]
- 接收<*,0>[0,1]和<*,0>[0,2]

- 通信开销$T = 6 * T_C$



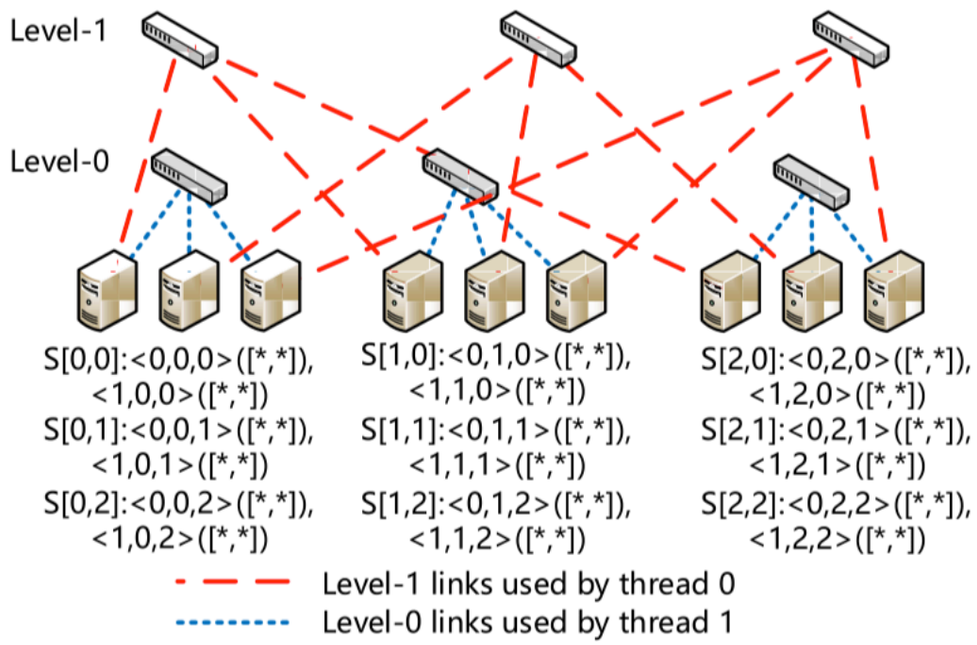
BML Design

Aggregation2

- 线程0:

- [0,0]传输<1,0>[0,*]给[1,0]、<2,0>[0,*]给[2,0]，接收[1,0]的<0,0>[1,*]和<0,0>[2,*]

- 通信开销$T = 2 * T_C$

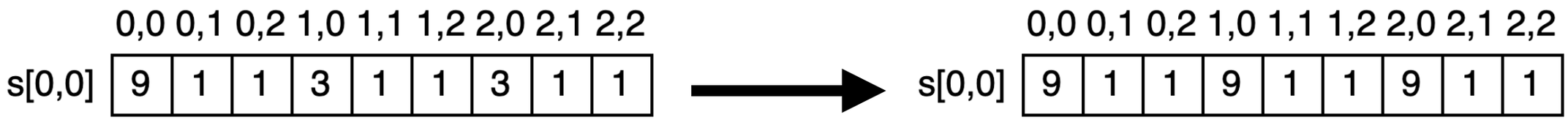


BML Design

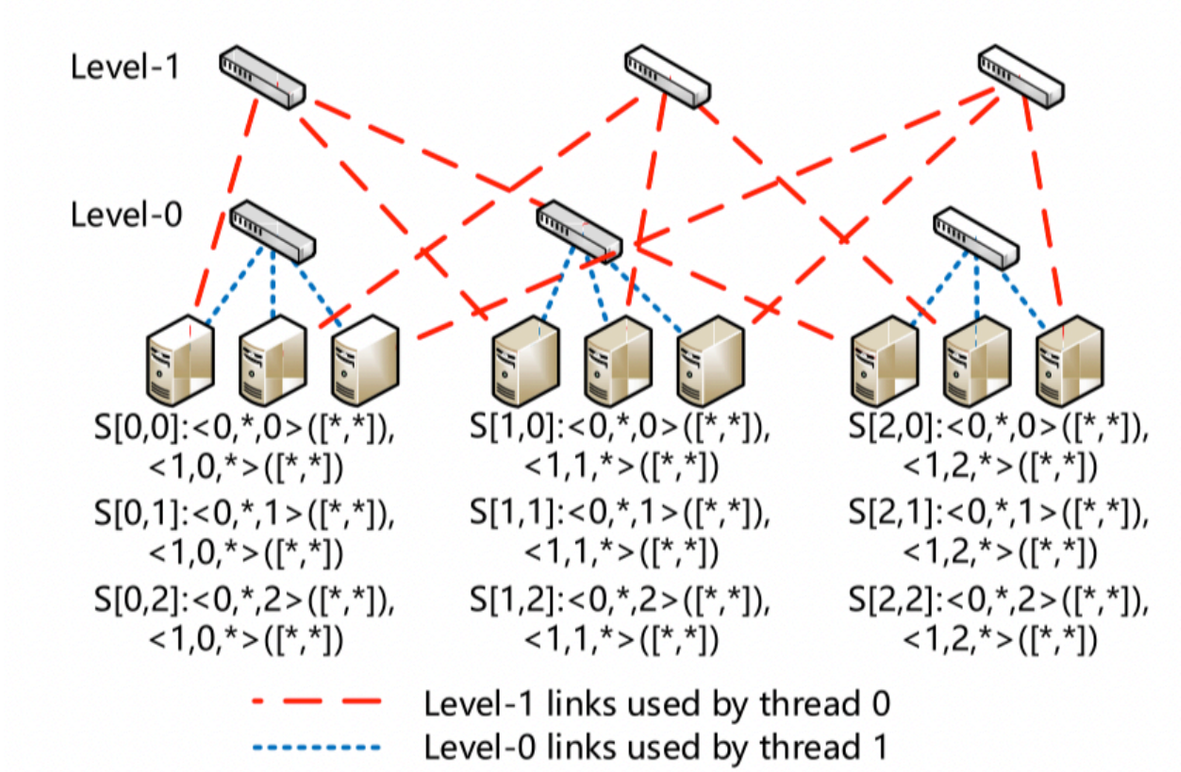
Broadcast1

- 线程0:

- [0,0]传输<0,0>[*,*]给[1,0]、[2,0]，接收<1,0>[*,*]和<2,0>[*,*]



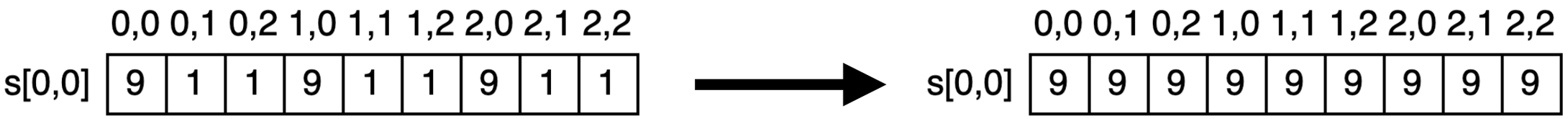
- 通信开销$T = 2 * T_C$



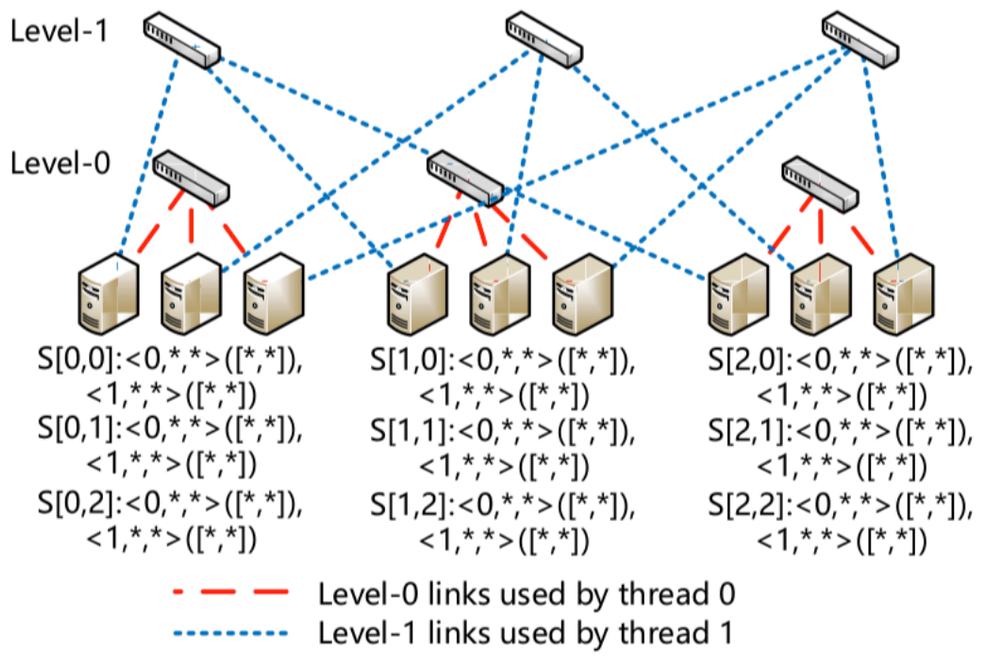
BML Design

Broadcast2

- 线程0:
- [0,0]将3个全聚合的梯度片段<*,0>[*,*]传递给[0,1]、[0,2]
- 同时接收[0,1]和[0,2]传递的三个全聚合的梯度片段<*,1>[*,*]、<*,2>[*,*]给[0,0]



- 通信开销$T = 6 * T_C$
- 总开销$T = 16T_C = \frac{8}{9}T_F$



BML Design

Fault Tolerance

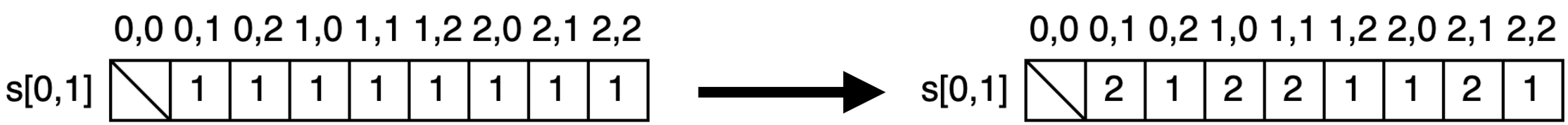
- Server Failure服务器失效
- Link Failure链路失效

- 计算调整：设x台服务器失效，原mini-batch为u，那么剩余$X-x$台机器的mini-batch调整为$u+\frac{u*x}{N-x}$
- 同步调整：全梯度信息划分成$k*(N-x)$片

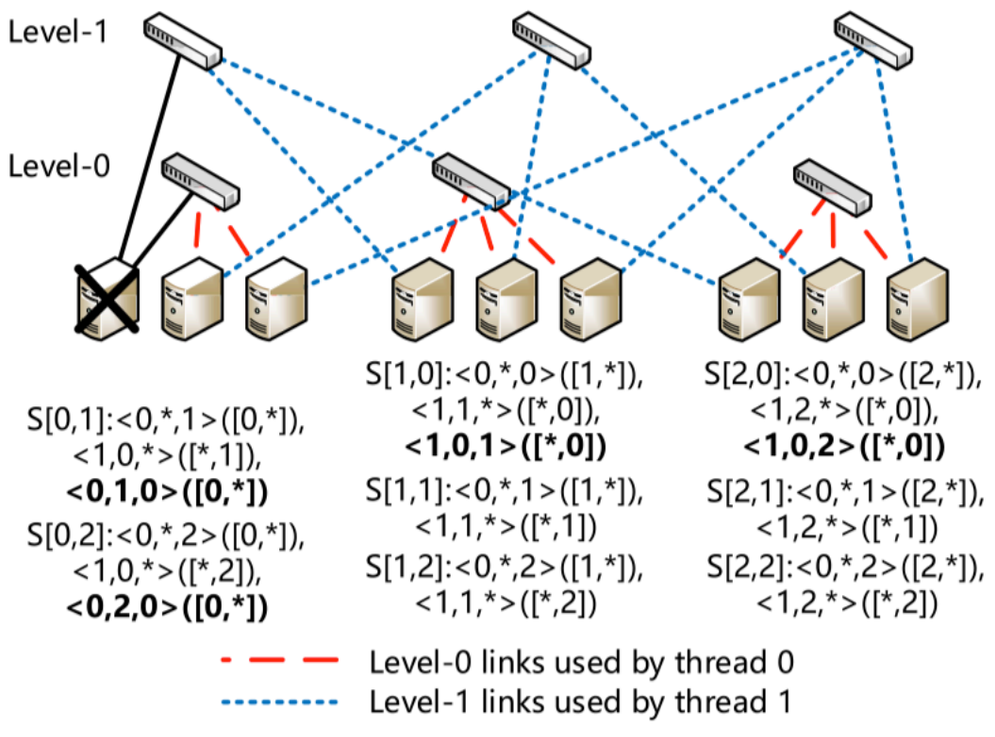
BML Design

Aggregation1

- [0,1]传输<*,2>和<0,1,0>给[0,2]，接收<0,*,1>[0,2]和<0,1,0>[0,2]



- 通信开销$T = 6 * T_C$

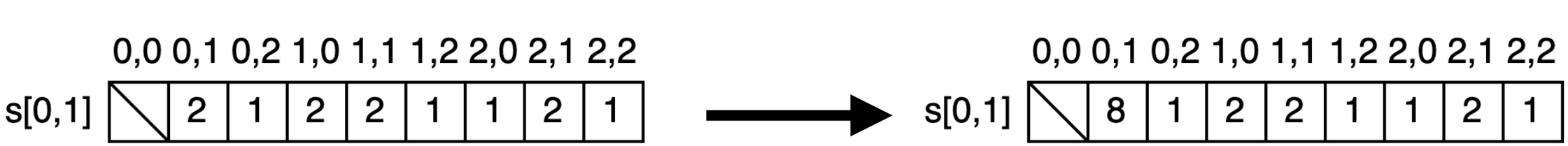


BML Design

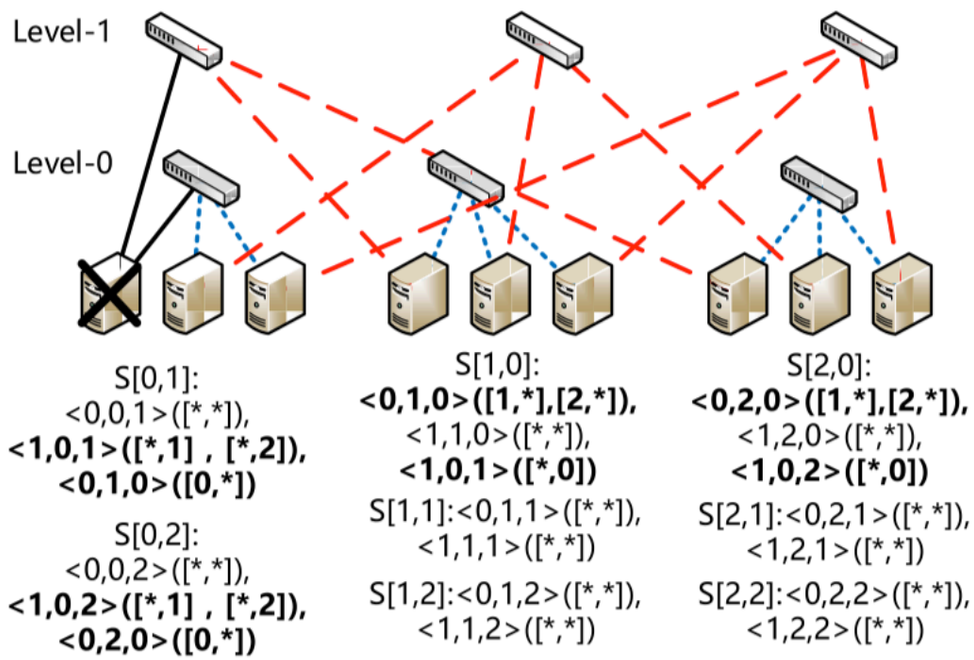
Aggregation2

- 线程0:

- [0,1]传输<0,1,1>[0,*]给[1,1]、<0,2,1>[0,*]给[2,1]，接收<0,1,0>[1,*]和<0,1,0>[2,*]



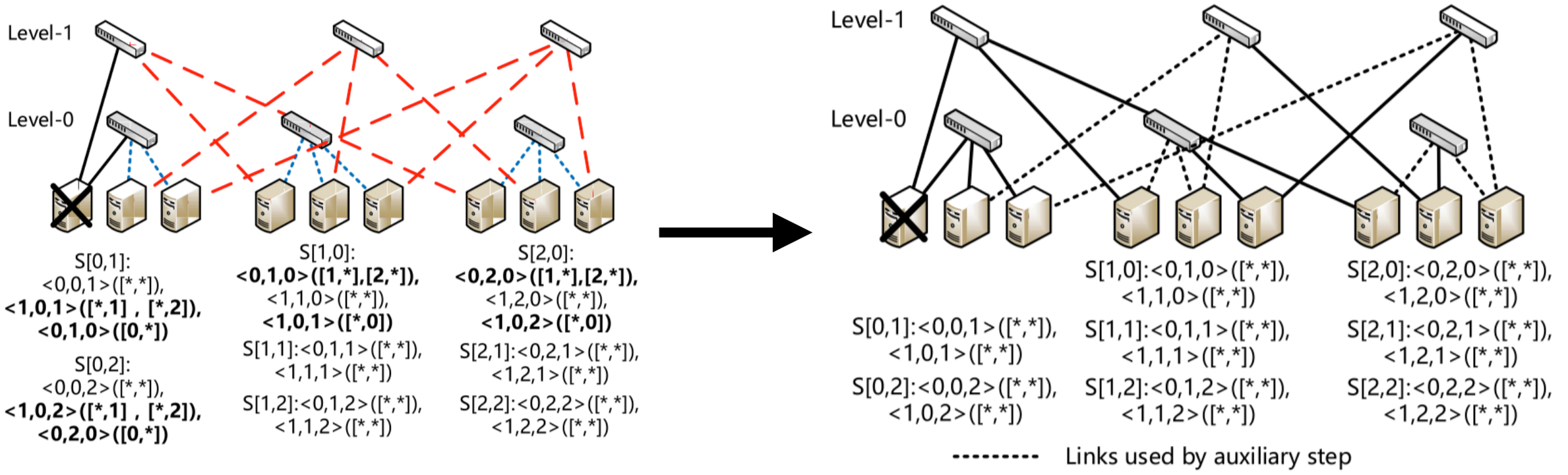
- 通信开销$T = 2 * T_C$



BML Design

Auxiliary2

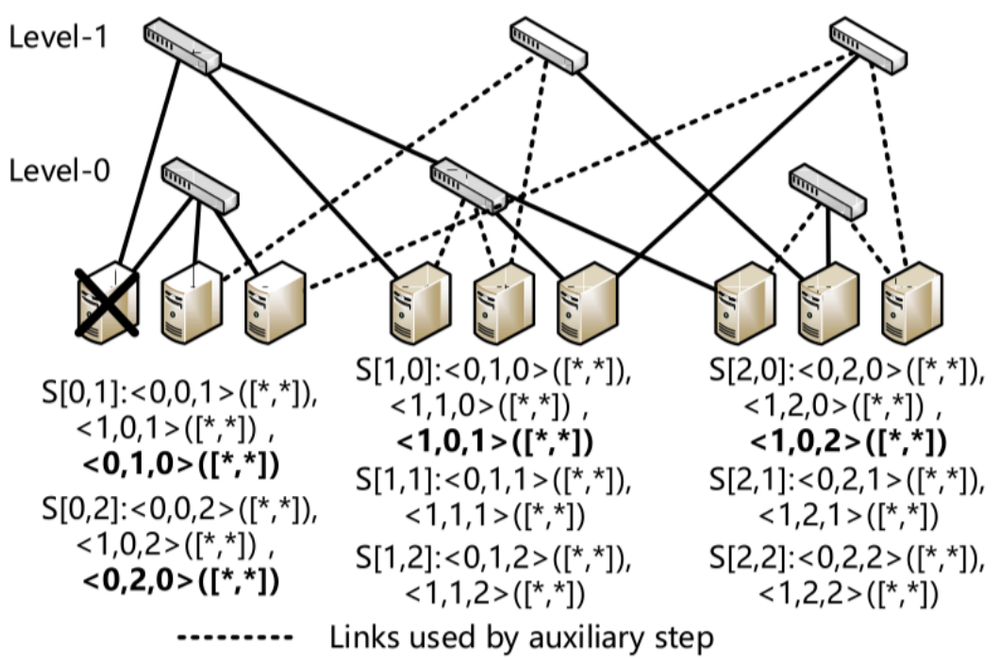
- 因为缺少[0,0]导致<0,1,0>[1,*]和<0,2,0>[2,*]无法进行同步
- [0,1]通过[0,1]->[1,1]->[1,0]传输<0,1,0>[0,*]给[1,0]
- [0,1]->[2,1]->[2,0]传输<0,1,0>[0,*]给[2,0]
- [0,0]接收<0,1,0>[1,*]和<0,1,0>[2,*]



BML Design

Auxiliary1

- [1,0]和[2,0]将汇聚后的<0,1,0>[*,*]传输给[1,0]
- [0,1]传输<0,1,0>[0,*]给[1,0]和[2,0]，接收<0,1,0>[1,*]和<1,0>[2,*]

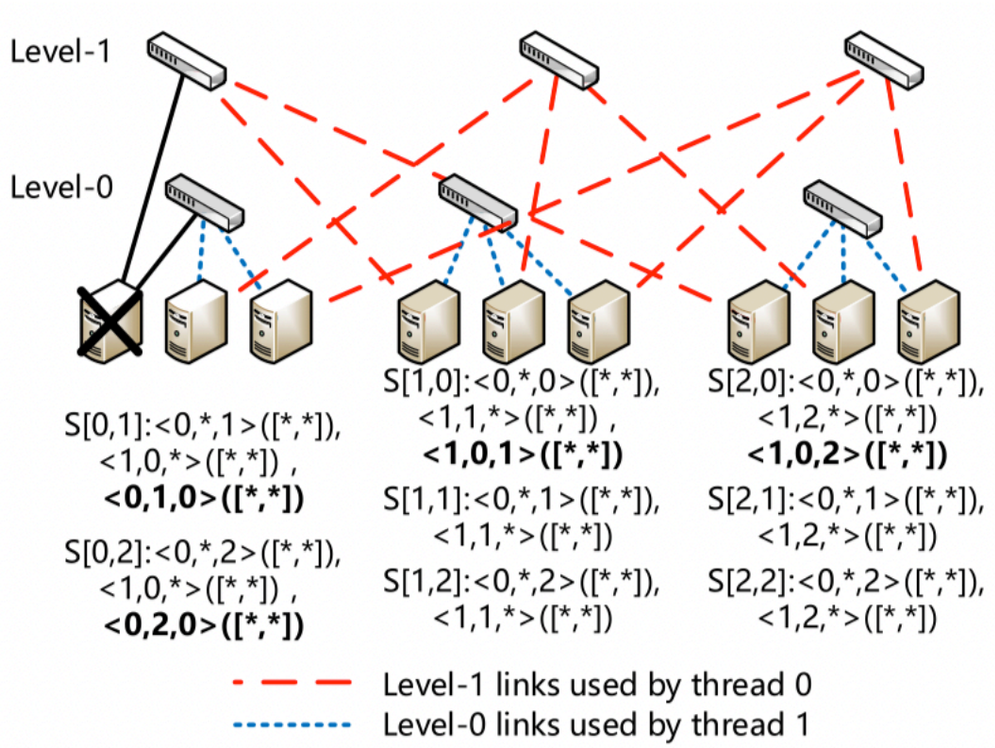


BML Design

Broadcast1

- [0,1]传输<0,0,1>[*,*]给[1,1]、[21]，接收<0,1,1>[*,*]和<0,2,1>[*,*]

- 通信开销$T = 2 * T_C$

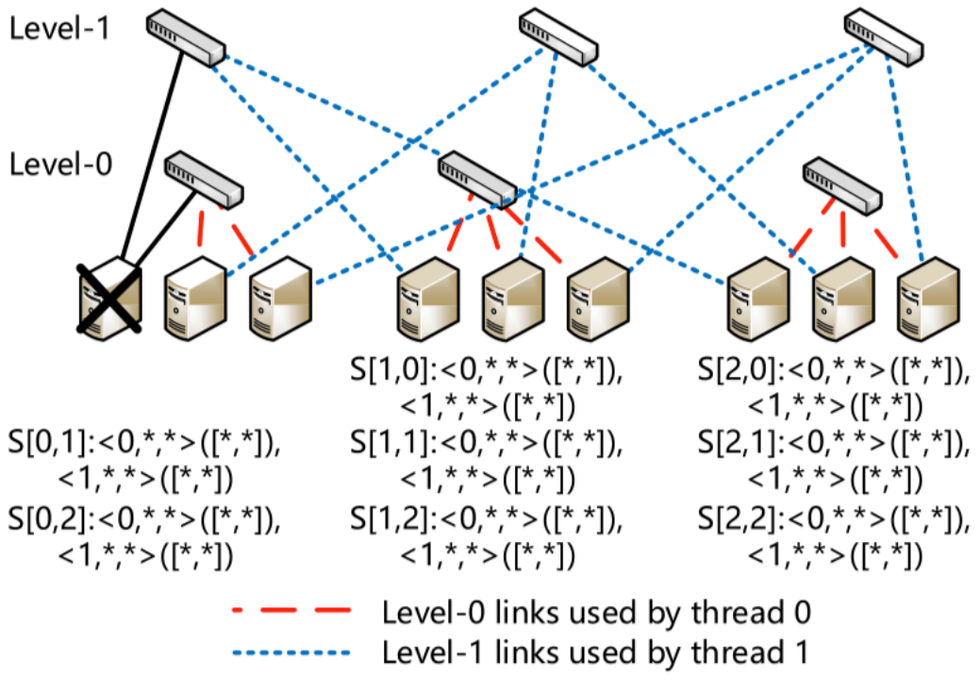


BML Design

Broadcast2

- [0,1]将全聚合的梯度片段<0,1,0>[*,*]传递给[0,2]
- [0,1]接收[0,2]传递的三个全聚合的梯度片段<*,2>[*,*]

- 通信开销$T = 6 * T_C$
- 总开销$T = 18T_C = \frac{9}{8}T_F$



BML Design

Link Failure链路失效

- 链路失效时，建议直接将链路上的服务器屏蔽，转变为Server Failure

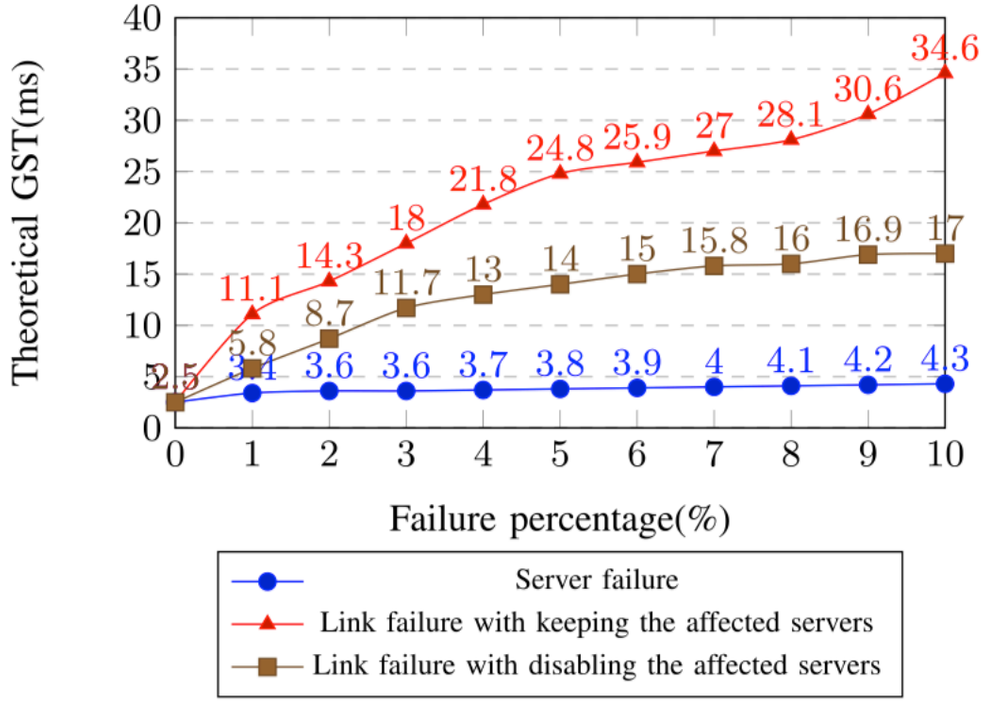
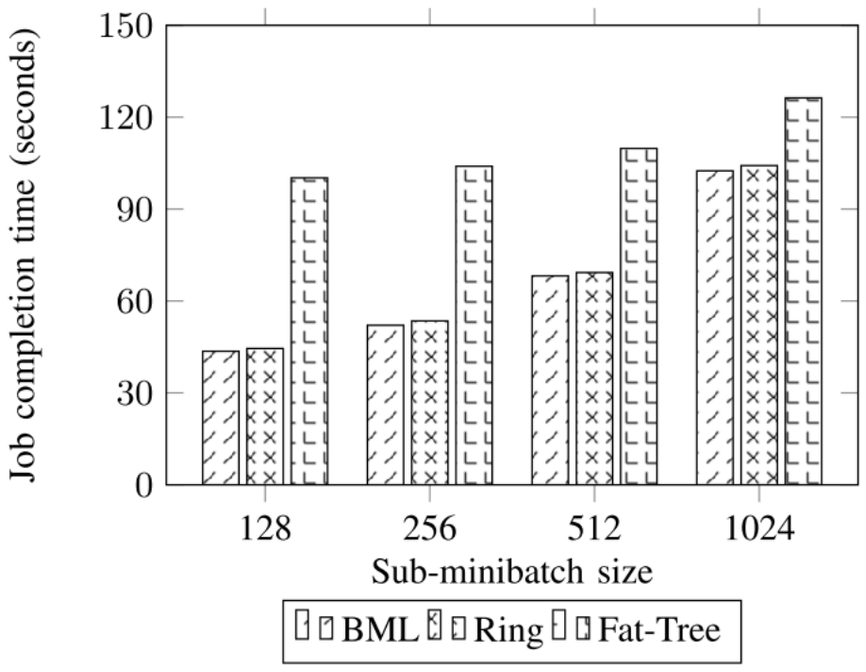


Fig. 19. Simulation results.

EXPERIMENTS

Result

- MNIST算法中，BML的完成时间比Fat-Tree快18.7%～56.4%
- Ring和BML相同完成时间，但是BML有容错机制，更有优势



EXPERIMENTS

Result

- VGG-19算法中，BML的完成时间比Fat-Tree快29.2%～52.1%
- Ring和BML完成时间相差2%，但是BML有容错机制，更有优势

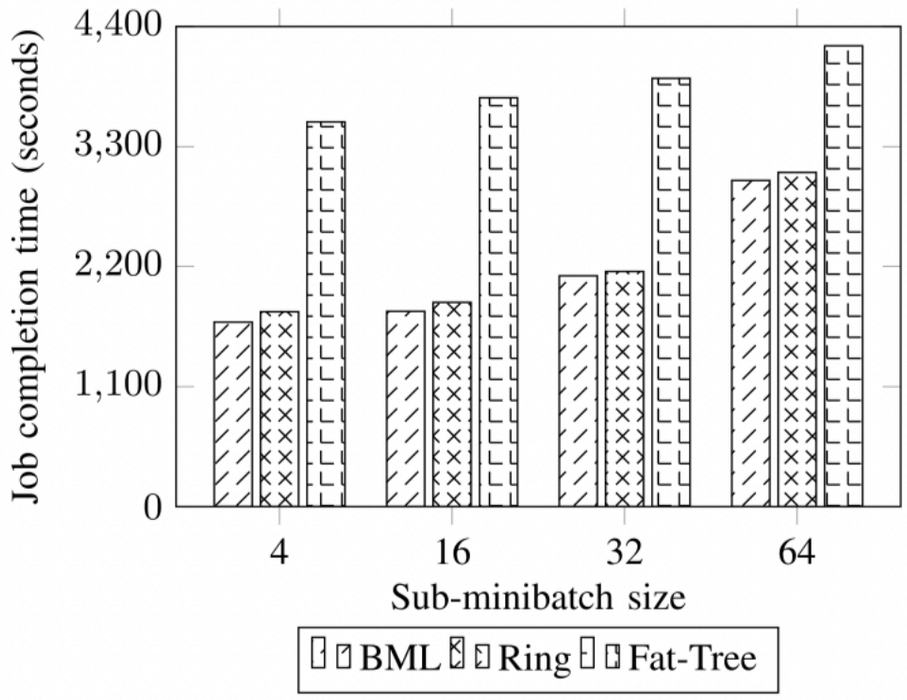


Fig. 21. Experiment result of VGG-19.

THINKING

- 没有单点故障的实验
- 没有分析拥塞对BML同步的影响