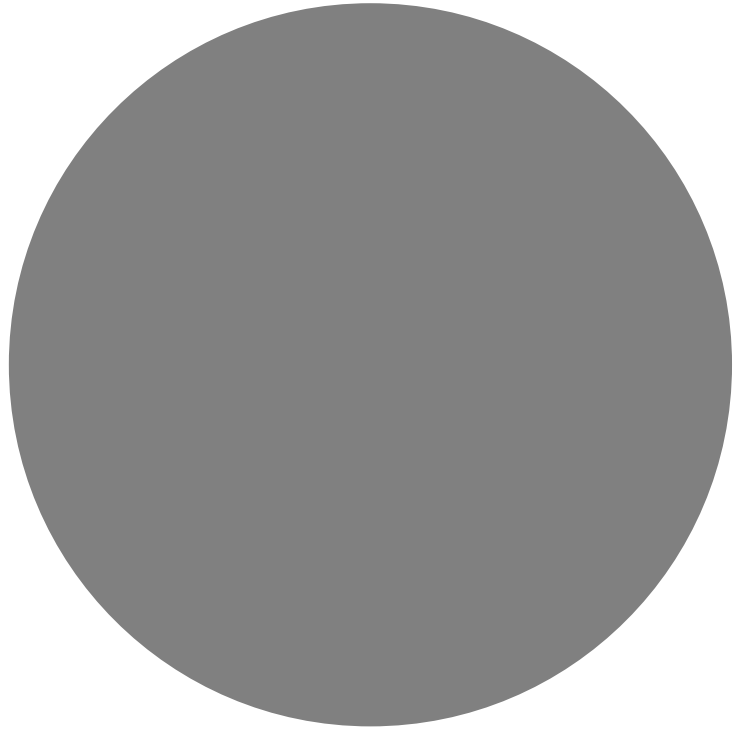
Toward Highly Available, Intelligent Cloud and ML Systems



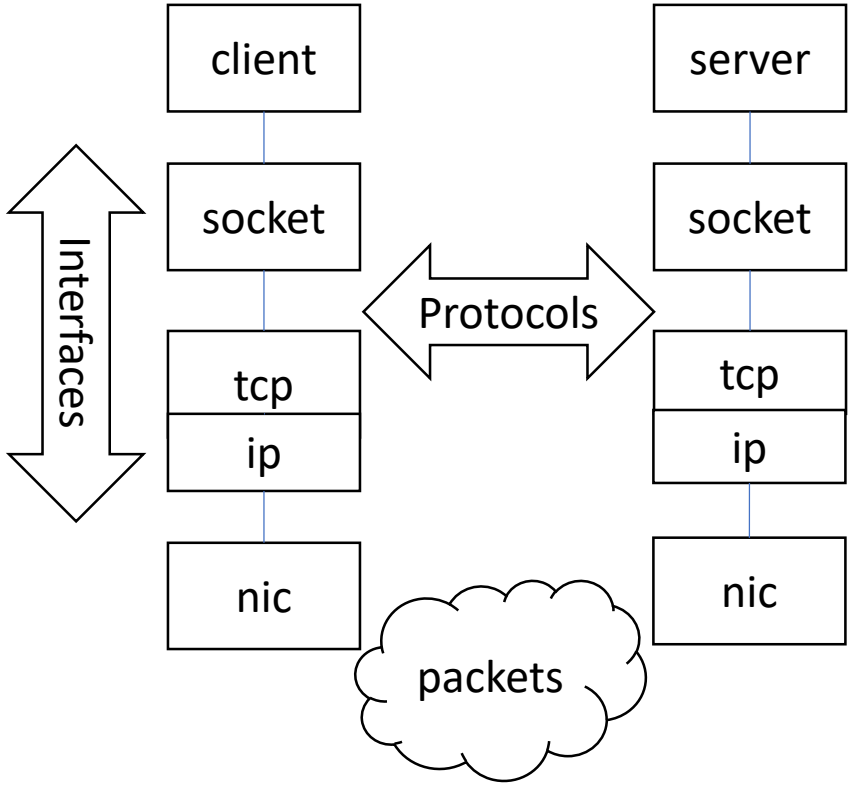
Chuanxiong Guo
Bytedance

NetAI 2018

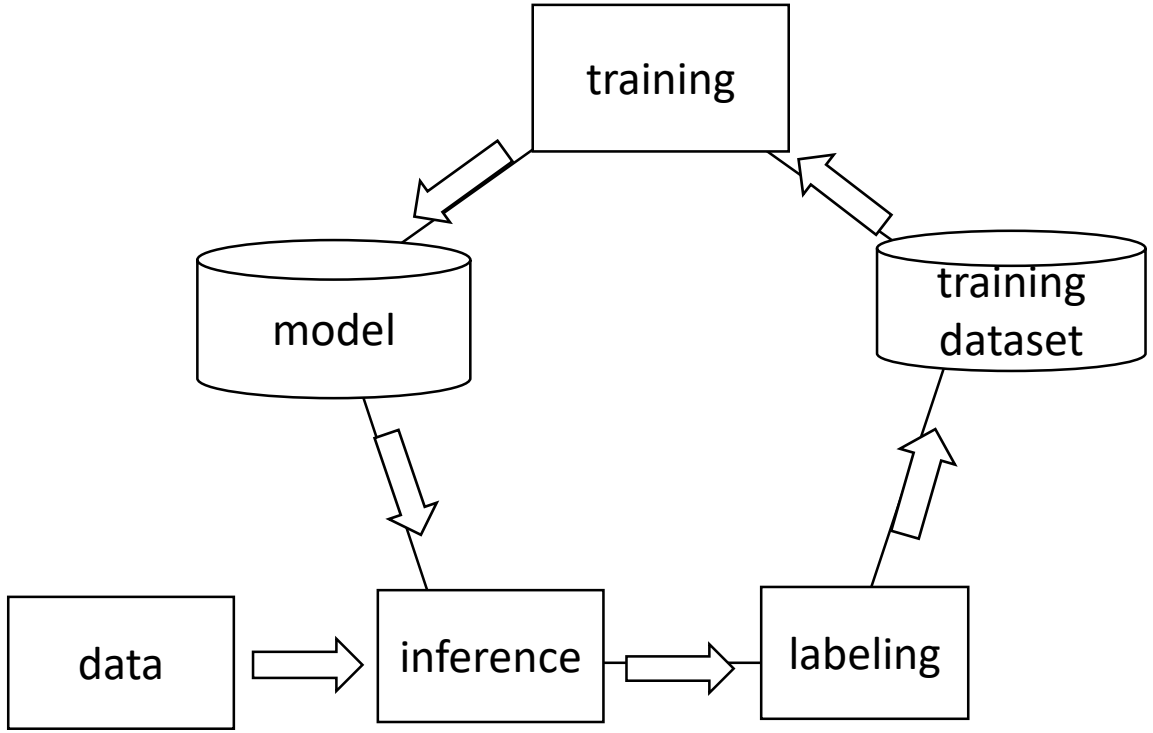
Outline

- Background: System/networking meets ML
- Deepview: ML for availability improvement of cloud systems
- RDMA for scalable ML training acceleration
- Summary

Two Different Approaches

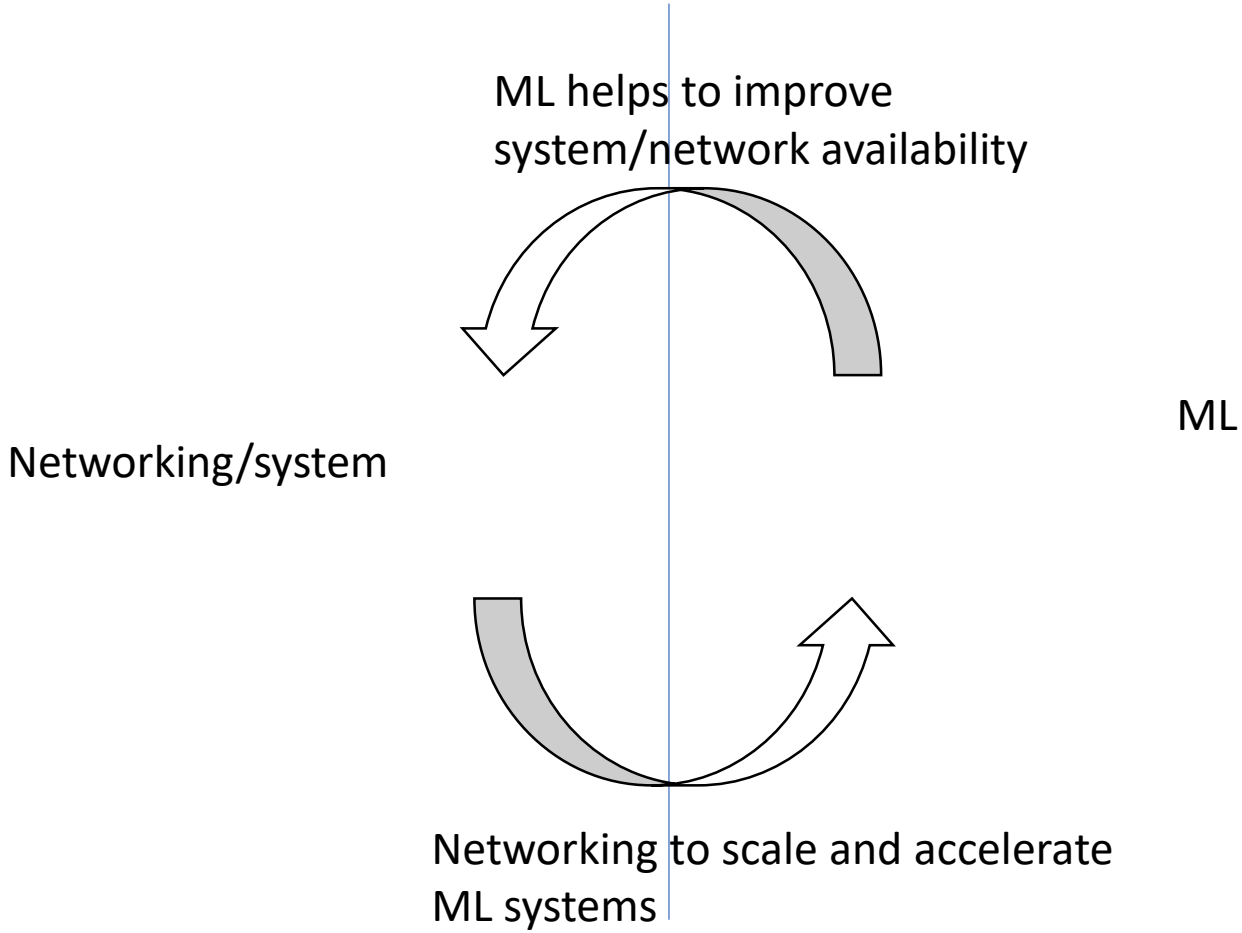


- Network/systems are *designed* by following *principles*
- Interfaces are explicitly defined, protocols are explicitly coded, and packets can be traced and explained

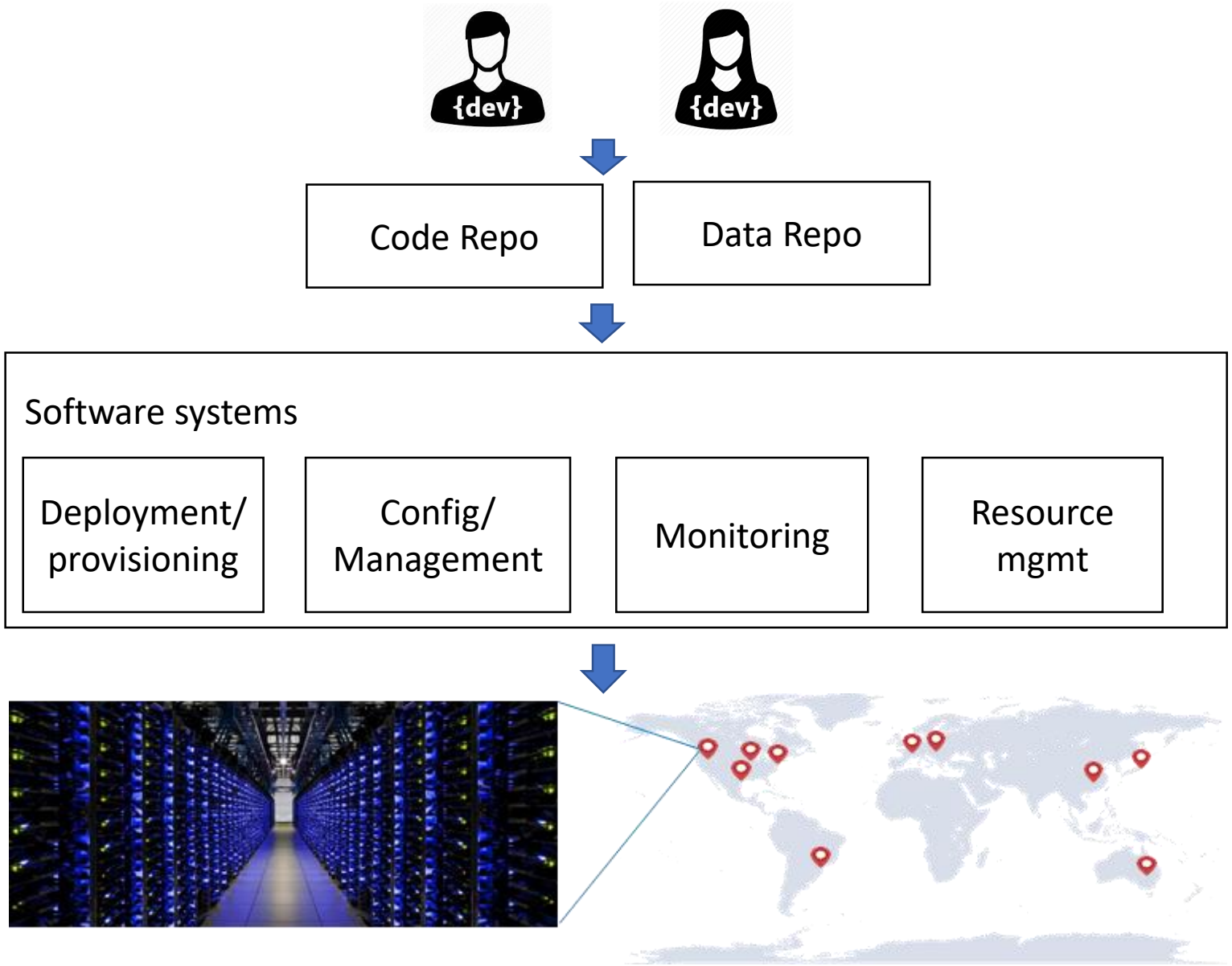


- Models in machine learning are *learned* from *data* without explicit programming
- Deep learning made breakthroughs in computer vision and speech

Networking Meets Machine Learning



Software Rules the Clouds



Incidents, Incidents, Incidents

Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region

We'd like to give you some additional information about the service disruption that occurred in the Northern Virginia (US-EAST-1) Region on the morning of February 28th. The Amazon Simple Storage Service (S3) team was debugging

Microsoft Azure Storage Issues Caused by Two Incidents

BY CHRIS BURT ON THURSDAY, MARCH 16 2017 ADD YOUR COMMENTS

Facebook is down in Asia-Pa… America, too (Update: It's ba…

Posted May 8, 2017 by Catherine Shu (@catherineshu)

6月27日阿里云故障说明

6月27日下午，我们在运维上的一个操作失误，导致一些客户访问阿里云官网控制台和使用部分产品功能出现问题，引发了大量吐槽。故障于北京时间2018年6月27日16:21左右开始，16:50分开始陆续恢复。

经过紧急技术复盘，故障原因如下：

当天下午，工程师团队在上线一个自动化运维新功能中，执行了一项变更验证操作。这一功能在测试环境验证中并未发生问题，上线到自动化运维系统后，触发了一个未知代码bug。错误代码禁用了部分内部IP，导致部分产品访问链路不通。 后续人工介入后，工程师团队快速定位问题进行了恢复。

受影响范围包括阿里云官网控制台，以及MQ、NAS、OSS等产品功能。

对于这次故障，没有借口，我们不能也不该出现这样的失误！我们将认真复盘改进自动化运维技术和发布验证流程，敬畏每一行代码，敬畏每一份托付。

阿里云计算有限公司
2018年6月27日

Feb 1, 2017 - GitLab

GitLab.com Database Incident

Yesterday we had a serious incident with one of our databases. We lost six hours of database data (issues, merge requests, users, comments, snippets, etc.) for GitLab.com.

Google Compute Engine Incident #16(

Connectivity issues in all regions

Incident began at **2016-04-11 18:25** and ended a… …ific).

DATE	TIME	DESCRIPTION
Apr 13, 2016	09:31	SUMMARY:
		On Monday, 11 A… …nces in all regions from 19:09 to 19

HPOU
@HPOUTX
Follow

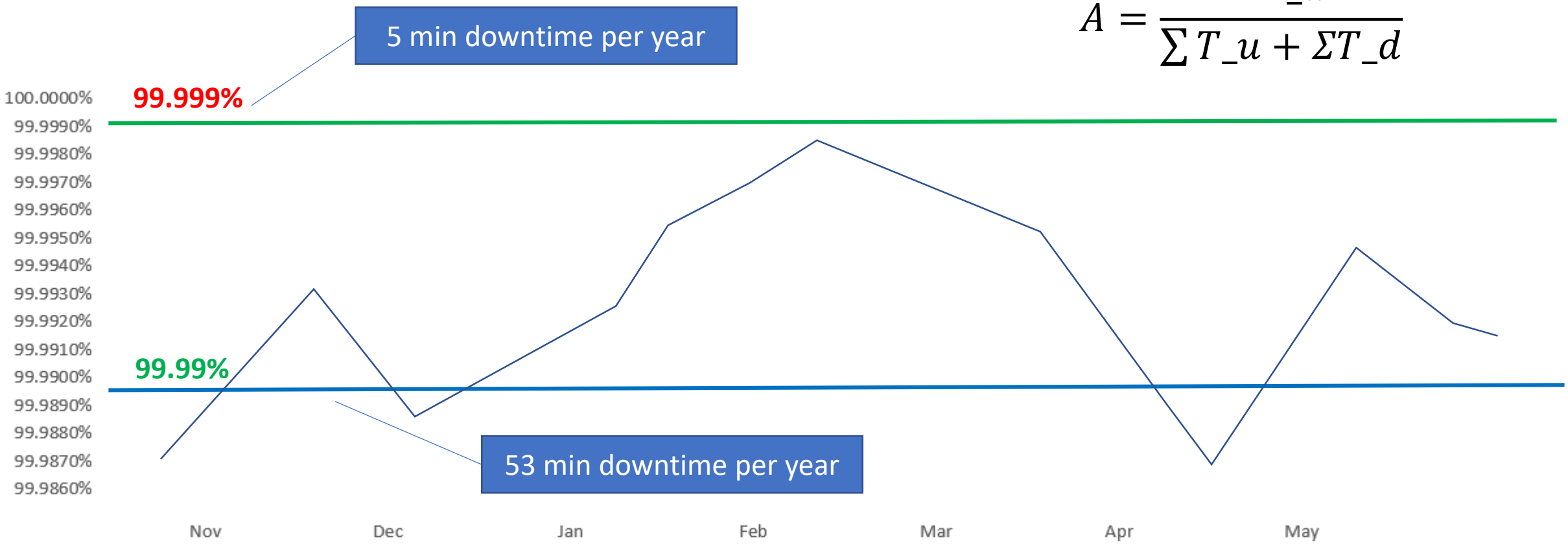
Folks please do not call the police because #facebookdown we are as upset as you are but we cannot fix facebook. #sorry #wetried #techpolice

RETWEETS 121 LIKES 69

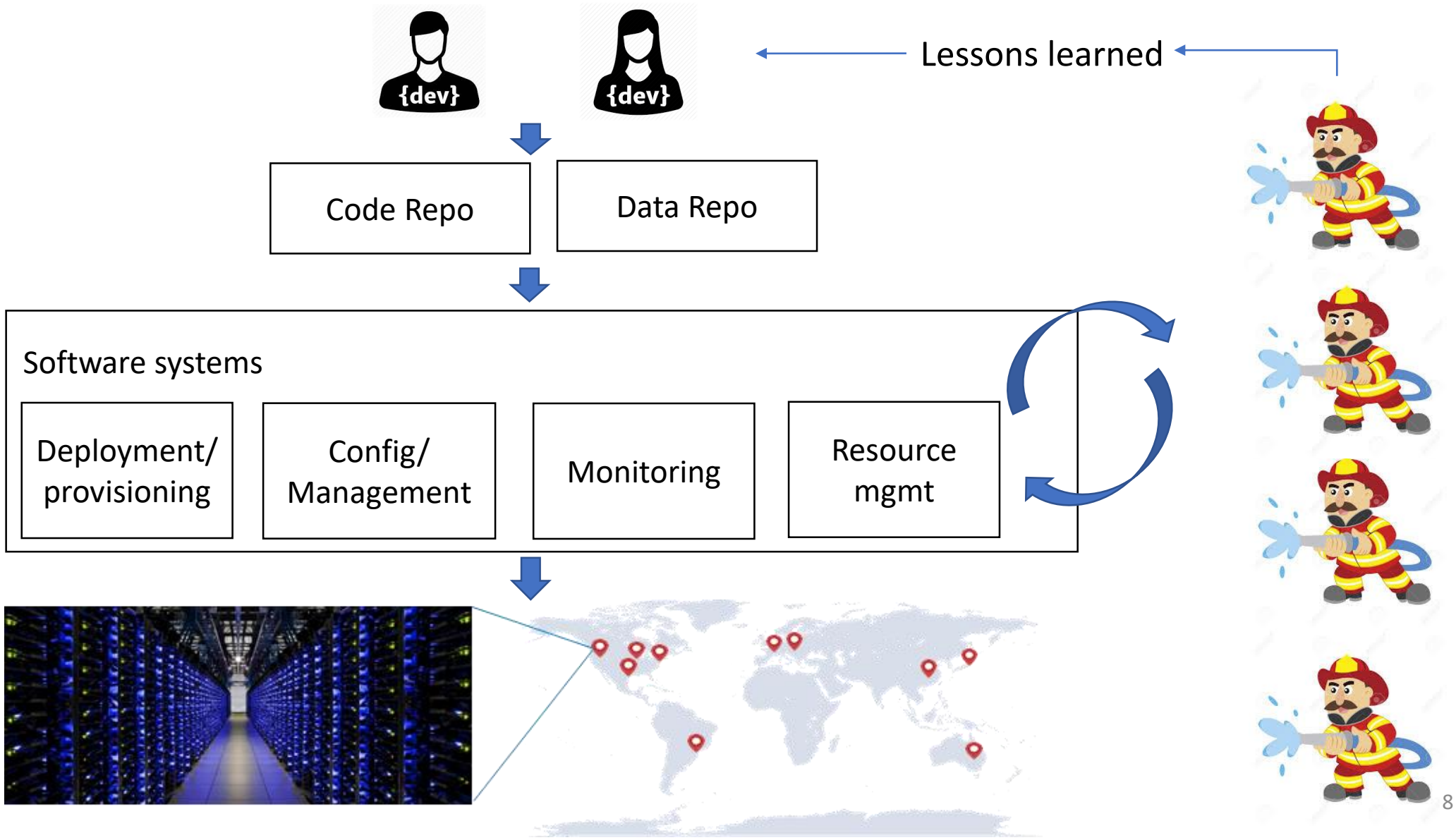
1:31 PM - 28 Sep 2015

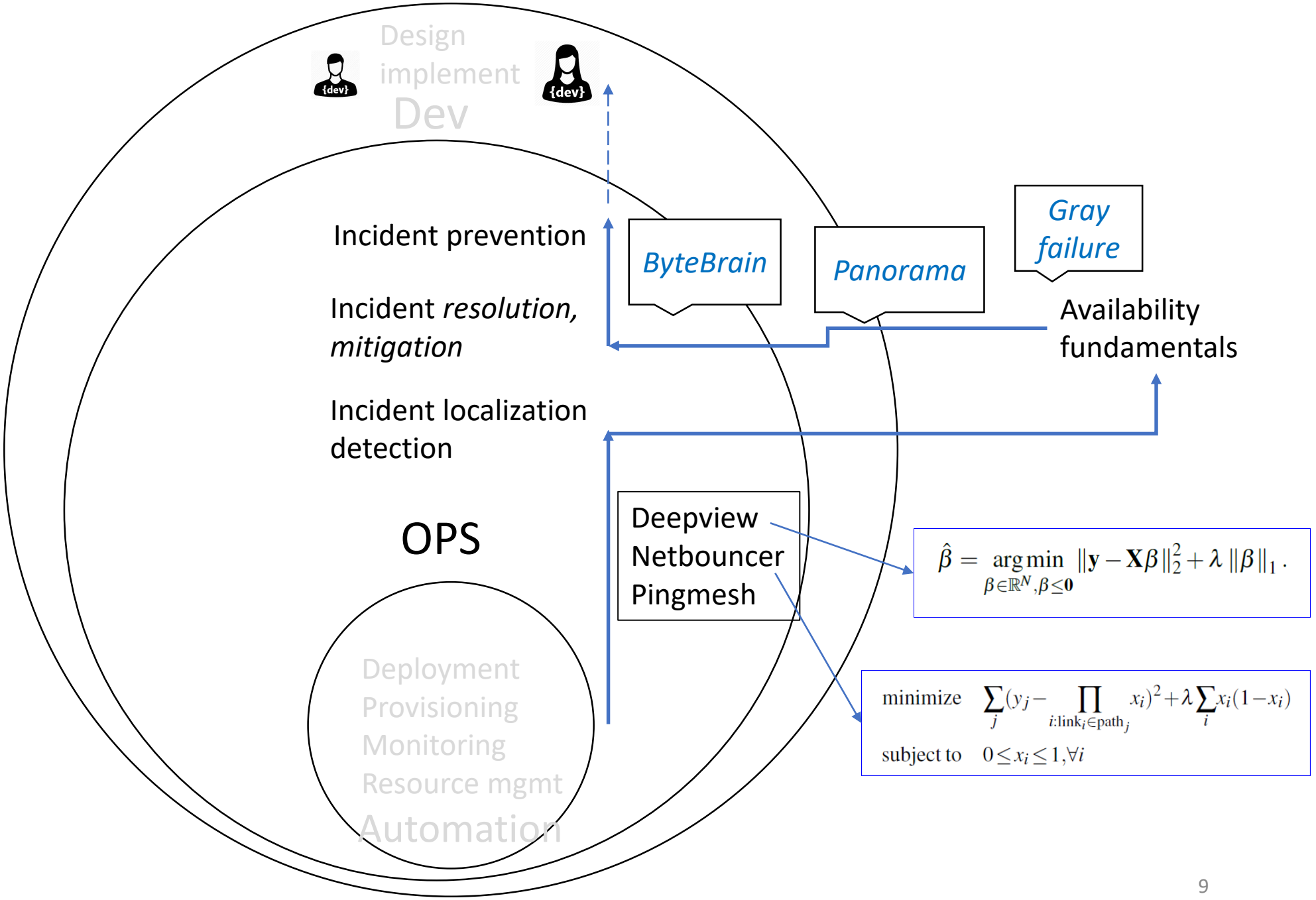
System Availability is Plagued by Incidents

$$A = \frac{\Sigma T\_u}{\sum T\_u + \Sigma T\_d}$$



Incident Handling Practice




Design
implement
Dev
Incident prevention
Incident *resolution, mitigation*
Incident localization detection
OPS
Deployment
Provisioning
Monitoring
Resource mgmt
Automation
ByteBrain
Panorama
Gray failure
Availability fundamentals
Deepview
Netbouncer
Pingmesh
$\hat{\beta} = \underset{\beta \in \mathbb{R}^N, \beta \leq \mathbf{0}}{\arg\min} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1 .$
minimize $\sum_j (y_j - \prod_{i:\text{link}_i \in \text{path}_j} x_i)^2 + \lambda \sum_i x_i(1-x_i)$
subject to $0 \leq x_i \leq 1, \forall i$

Deepview for Virtual Disk Failure Diagnosis

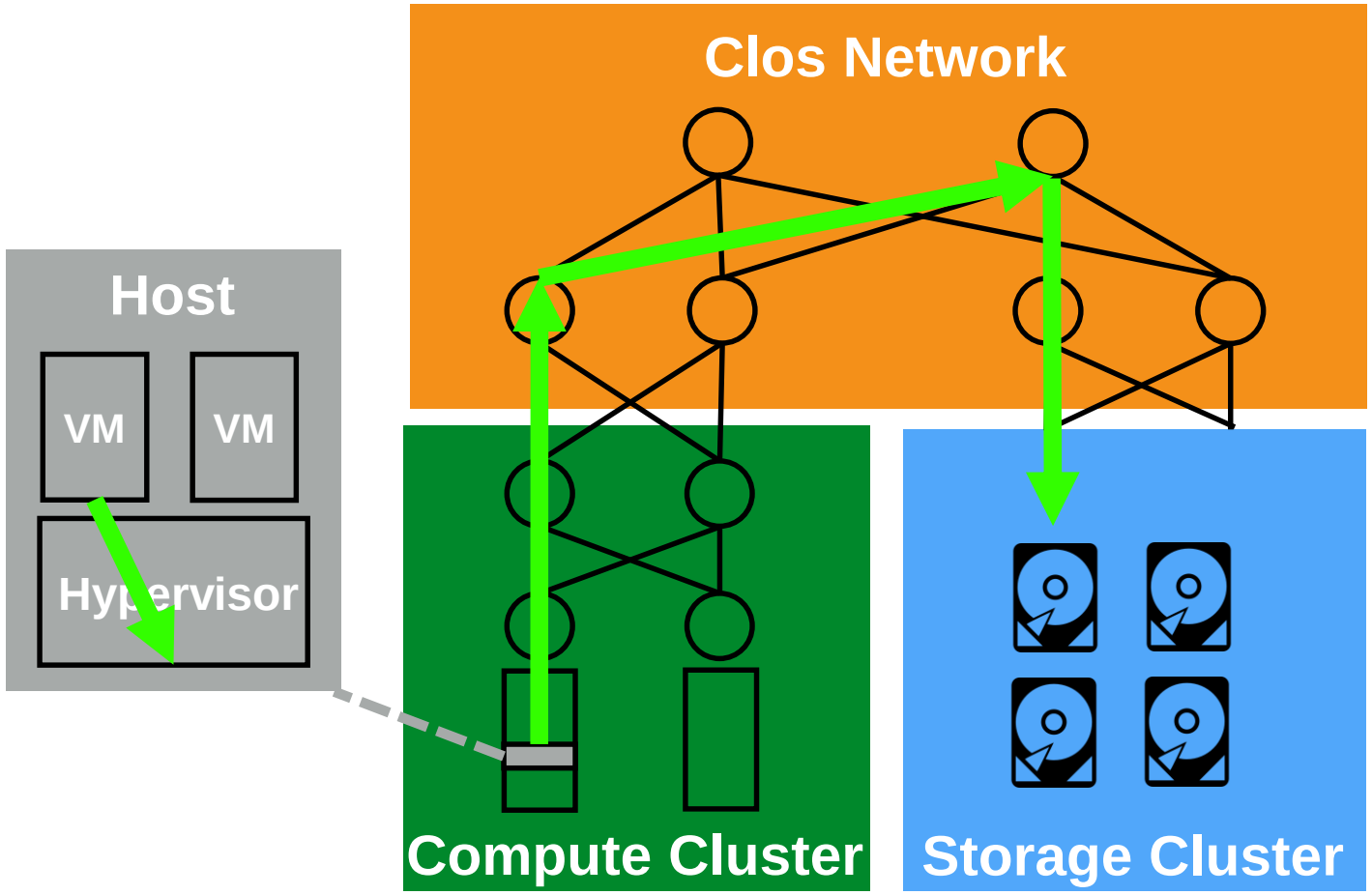
-- A case where ML helps system availability

VM Availability

- IaaS is one of the largest cloud services today
- High VM availability is a key performance metric
- Yet, achieving 99.999% VM uptime remains a challenge

1. **What is the VM availability bottleneck?**
2. **How to eliminate it?**

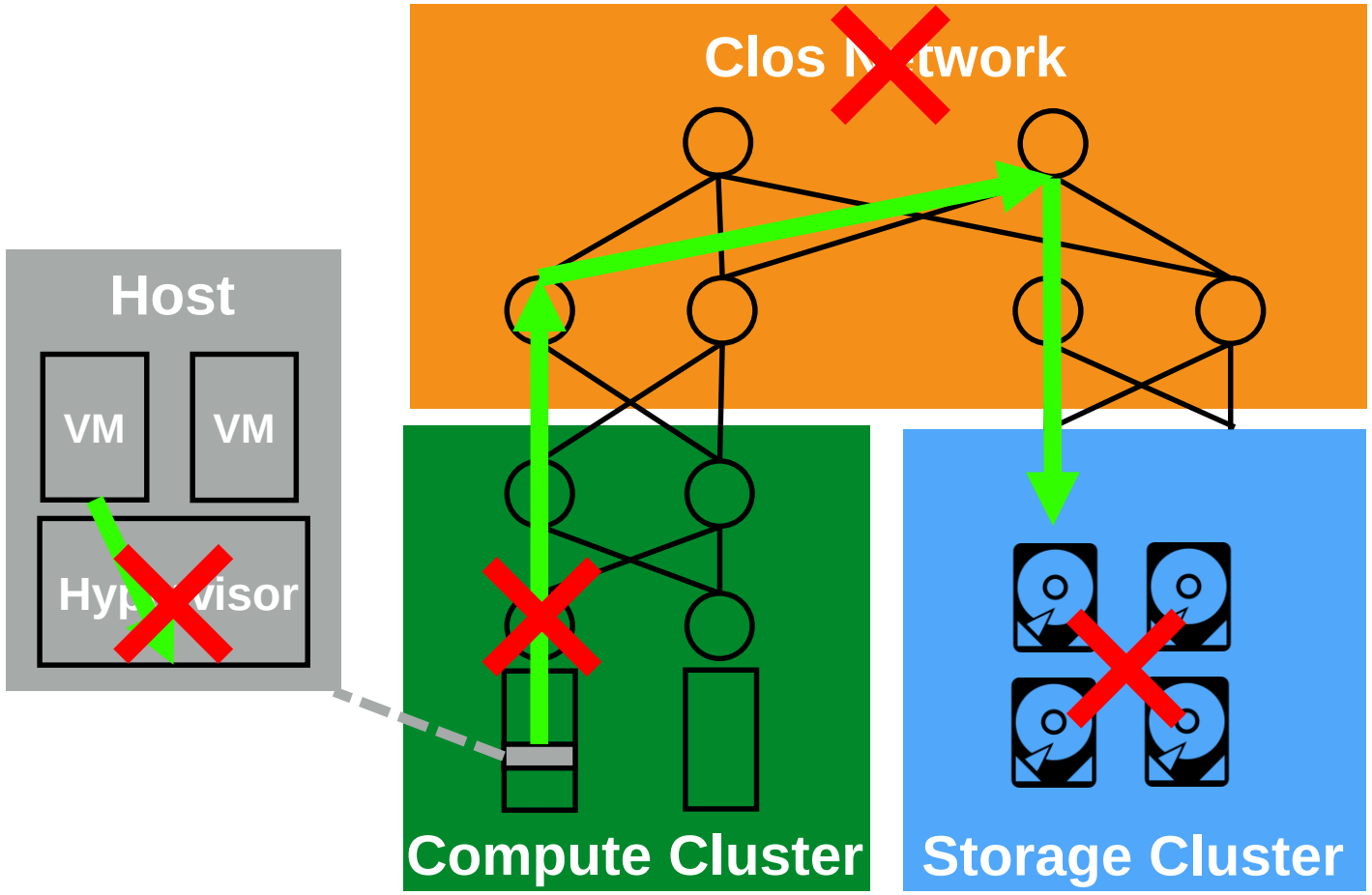
IaaS Architecture



Subsystems inside a Datacenter

- Compute and storage clusters with a Clos-like network
- **Compute-storage Separation**
  - VMs and Virtual Hard Disks (VHDs) provisioned from different clusters
  - Hypervisor transparently redirects disk access to remote storage
- Keep data available during localized power failure to a rack

A New Type of Failure: VHD Failures

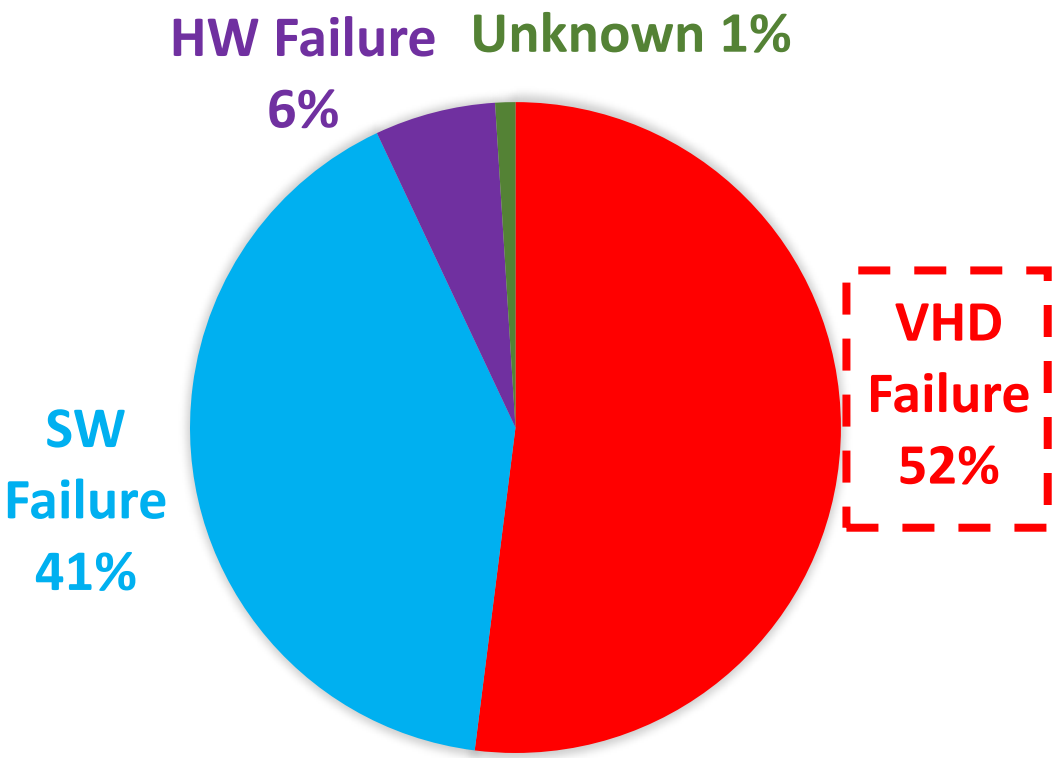


Subsystems inside a Datacenter

- Infra failures can disrupt VHD access
- Hypervisor can retry, but not indefinitely
- Hypervisor will **crash the VM** to surface failures to customer
- Allow customers to take actions to keep their app-level SLAs

How much do VHD failures impact VM availability?

Availability Bottleneck



Breakdown of Unplanned VM Downtime in a Year

- **VHD failure localization is the bottleneck**
  - **52%** of unplanned VM downtime
  - Take 10s minutes to hours to localize
- This talk: quick and accurate failure localization

Failure Triage was Slow and Inaccurate

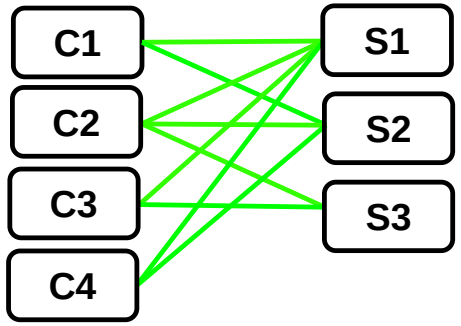
- SREs from each team check their subsystem for anomalies to match the incident
  - e.g. compute host heart-beats, storage perf-counters, network link discards

- Incidents get ping-ponged among different teams due to false positives
  - Inaccurate diagnosis and delayed mitigation

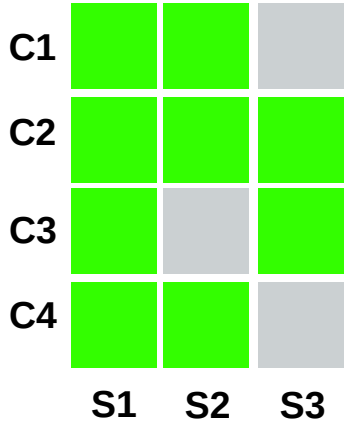
- Gray failures in network and storage are hard to catch
  - Troubled but not totally down, e.g. performance issues or software bugs
  - Only fail a subset of VHDs requests
  - Can take hours to localize

Deepview Approach: Global View

- Isolate failures by examining interactions between subsystems
  - Instead of alerting every SRE team to check if their subsystem is at fault
- Bipartite model
  - Compute Clusters (left) : Storage Clusters (right)
  - VMs are provisioned from compute/storage cluster pair
  - Edge weight = VHD failure rate



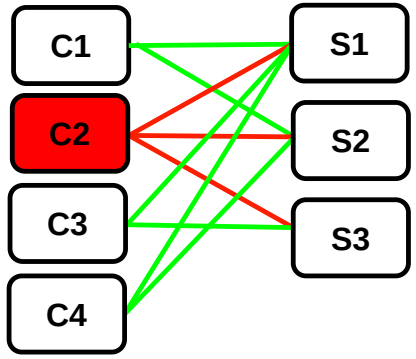
Bipartite Model



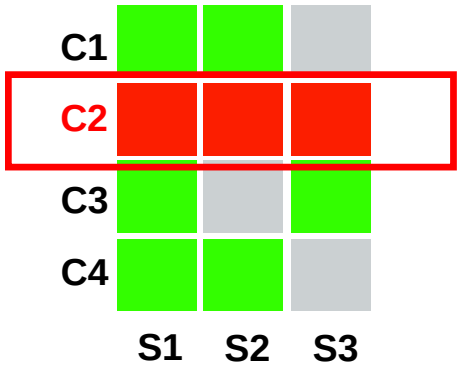
Grid View

Our Approach: Global View

Example Compute Cluster Failure

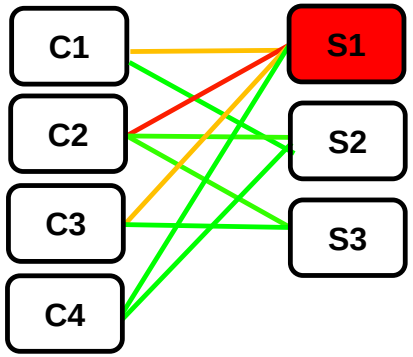


Compute Cluster
C2 failed

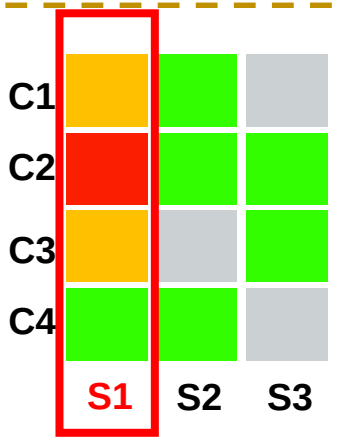


C2 Failure
Grid View

Example Storage Cluster Failure



Storage Cluster
S1 Failed



S1 Gray Failure
Grid View

Challenges

Remaining challenges:

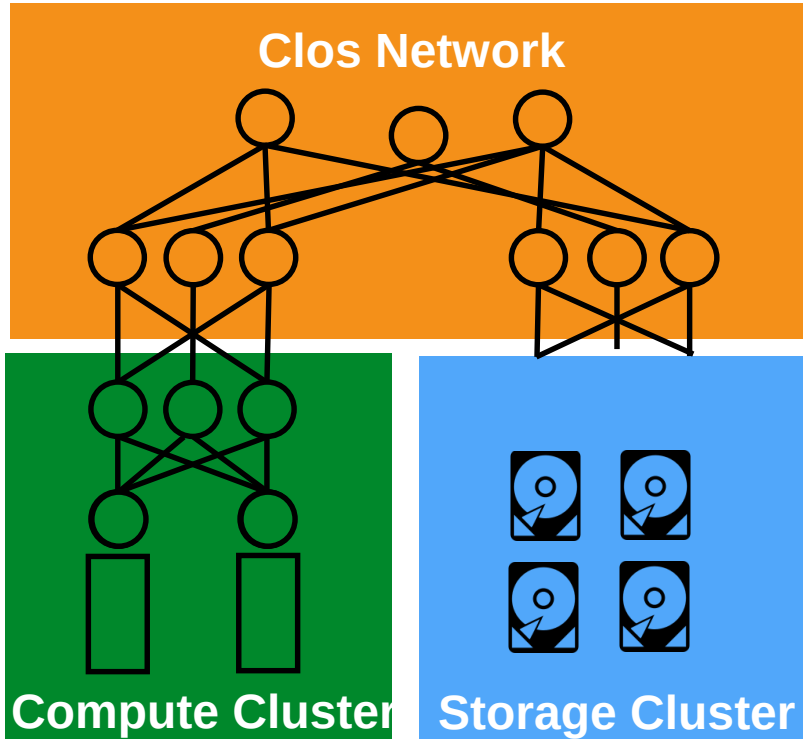
1. Need to pinpoint network failures → Generalized model to include network devices
2. Need to handle gray failures → Lasso regression/Hypothesis testing algorithm
3. Need to be near-real-time → Streaming data pipeline

Summary of our goal:

A system to localize VHD failures to underlying failures in compute, storage or network subsystems within a time budget of 15 minutes

Time budget set by production team to meet availability goals

Deepview Model: Include the Network



- Need to handle multipath and ECMP
- Simplify Clos network to a tree by aggregating network devices
- Can model at the granularity of clusters or ToRs

Deepview Model: Estimate Component Health

$$\mathbf{Prob(path\ i\ is\ healthy)} = \prod_{\mathbf{j \in path(i)}} \mathbf{Prob(component\ j\ is\ healthy)}$$

Blue: observable
Red: unknown
Purple: topology

$$\mathbf{1 - \frac{e_i}{n_i} = \prod_{j \in path(i)} p_j}$$

*Assume independent failures

$\mathbf{e_i}$=num of VMs crashed
$\boldsymbol{n_i}$=num of VMs

$$\mathbf{\log\left(1 - \frac{e_i}{n_i}\right) = \sum_{j \in path(i)} \log p_j}$$

System of Linear Equations

Component j is healthy with
$\mathbf{p_j = \exp(\beta_j)}$
- $\beta_j = 0$, clear component j
- $\beta_j \ll 0$, may blame it

$$\mathbf{y_i = \sum_{j=1}^{N} \beta_j\ x_{ij} + \varepsilon_i}$$

$\mathbf{y_i}$=$\mathbf{\log\left(1 - \frac{e_i}{n_i}\right)}$
$\boldsymbol{\beta_j}$=$\mathbf{\log p_j}$
$\boldsymbol{\varepsilon_i}$=measurement noise

Deepview Algorithm: Prefer Simpler Explanation via Lasso

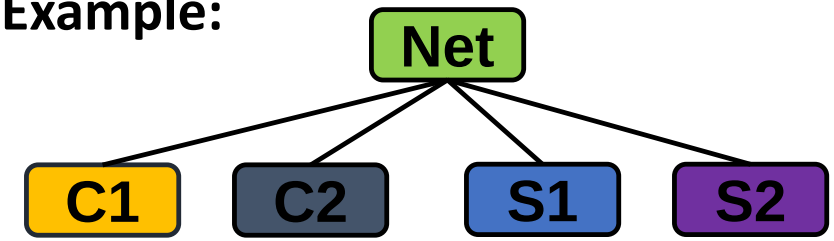
$$y_i = \sum_{j=1}^{N} \beta_j \, x_{ij} + \varepsilon_i$$

- Potentially #unknowns > #equations
- Traditional least-square regression would fail

- But multiple simultaneous failures are rare
- **How to encode this domain knowledge mathematically?**

- Equivalent to prefer most β_j to be zero
- **Lasso regression** can get sparse solutions efficiently

Example:



$$y_1 = \beta_{c1} + \beta_{net} + \beta_{s1} + \varepsilon_1$$
$$y_2 = \beta_{c1} + \beta_{net} + \beta_{s2} + \varepsilon_2$$
$$y_3 = \beta_{c2} + \beta_{net} + \beta_{s1} + \varepsilon_3$$
$$y_4 = \beta_{c2} + \beta_{net} + \beta_{s2} + \varepsilon_4$$

Lasso Objective Function:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^N, \beta \leq 0}{\operatorname{argmin}} \|y - X\beta\|^2 + \lambda \|\beta\|_1$$

Sparsity

Deepview Algorithm: Principled Blame Decision via Hypothesis Testing

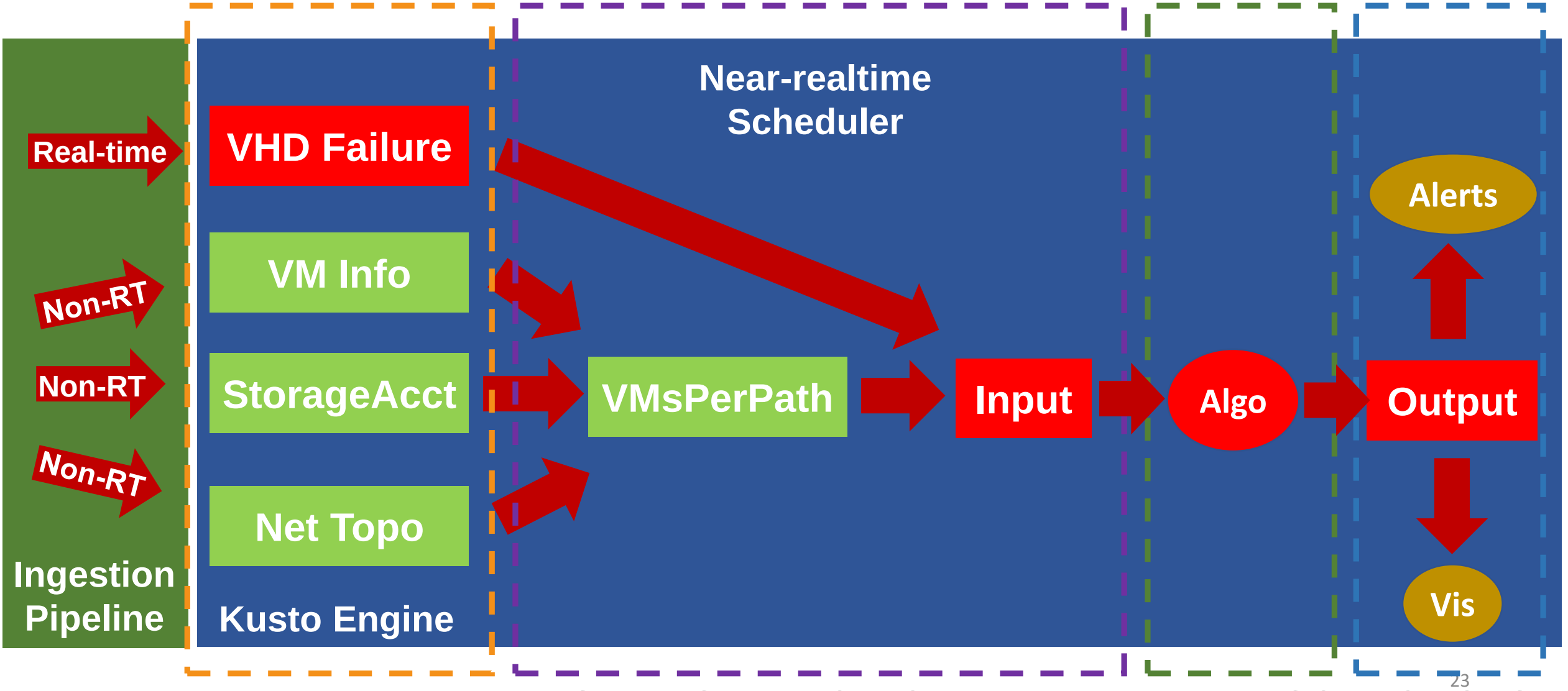
- Need a binary decision (**flag/clear**) for each component
- Ad-hoc thresholds do not work reliably
- **Can we make a principled decision?**

- *If estimated failure probability worse than average, then likely a real failure*
- Automate this empirical decision criterion using a hypothesis test:

$$\mathbf{H_0(j)}: \boldsymbol{\beta_j = \overline{\beta}} \quad \textbf{vs.} \quad \mathbf{H_A(j)}: \boldsymbol{\beta_j < \overline{\beta}}$$

- Reject $\mathrm{H_0(j)}$ means blame component j
- Otherwise, clear component j

Deepview System Architecture: NRT Data Pipeline



Some Statistics

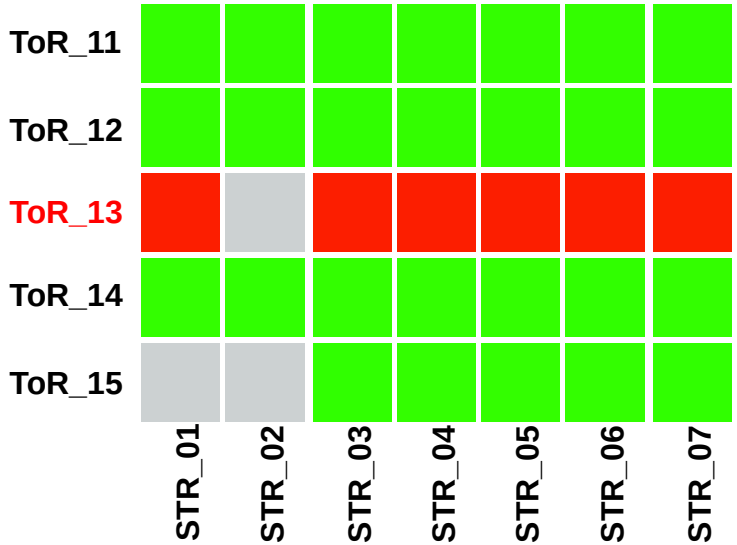
- Analyzed Deepview results for one month
  - Daily VHD failures: hundreds to tens of thousands
- Detected 100 failures instances
  - 70 matched with existing tickets, 30 were previously undetected
- Reduced unclassified VHD failures to less than a max of 500 per day
  - Single-host failures or customer mistakes (e.g. expired storage accounts)

Case Study 1: Unplanned ToR Reboot

- Unplanned ToR reboot can cause VMs to crash
- We knew this can happens, but not where and when

- Deepview can flag those ToRs
- The figure shows a ToR down in one small region
  - Blamed the right ToR among 288 components

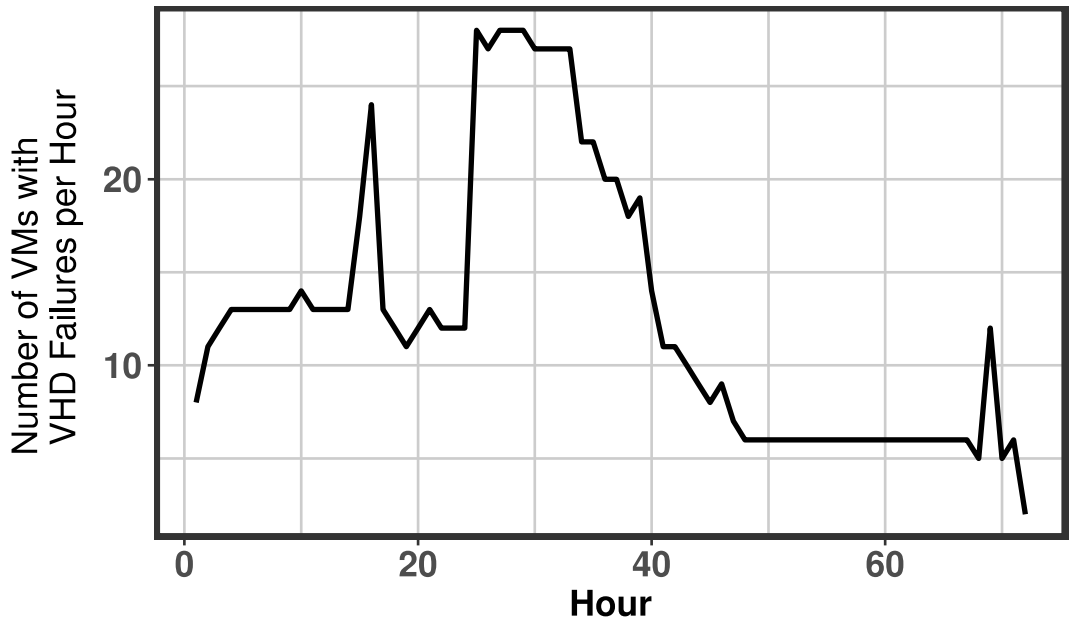
- Associate VM downtime with ToR failures
- Quantify the impact of ToR as a single-point-of-failure on VM availability



Unplanned ToR reboot in a region

Case Study 2: Storage Cluster Gray Failure

- Impact only a subset of VMs
- A storage cluster was brought online with a bug that puts some VHDs in negative cache
- Deepview flagged the faulty storage cluster almost immediately while manual triage took 20+ hours



Number of VMs with VHD Failures per Hour during a Storage Cluster Gray Failure

Deepview Insight: ToR as a Single Point of Failure

- **Reduced Network Cost vs. Availability cost for using a single ToR per rack**
- Unplanned ToR failures: soft failures (recoverable by reboot) vs. hard failures

ToR Availability

$$= 1 - \frac{(\%\ \mathbf{soft} * \mathbf{soft\ dur.} + \%\ \mathbf{hard} * \mathbf{hard\ dur.}) * \mathbf{frac.\ rebooted\ ToRs\ per\ month}}{\mathbf{total\ time\ in\ a\ month}}$$

$$= 1 - \frac{(90\% * \mathbf{20\ min} + 10\% * \mathbf{120\ min}) * 0.1\%}{\mathbf{30} * \mathbf{24} * \mathbf{60\ min}}$$

$$= 99.99993\%$$

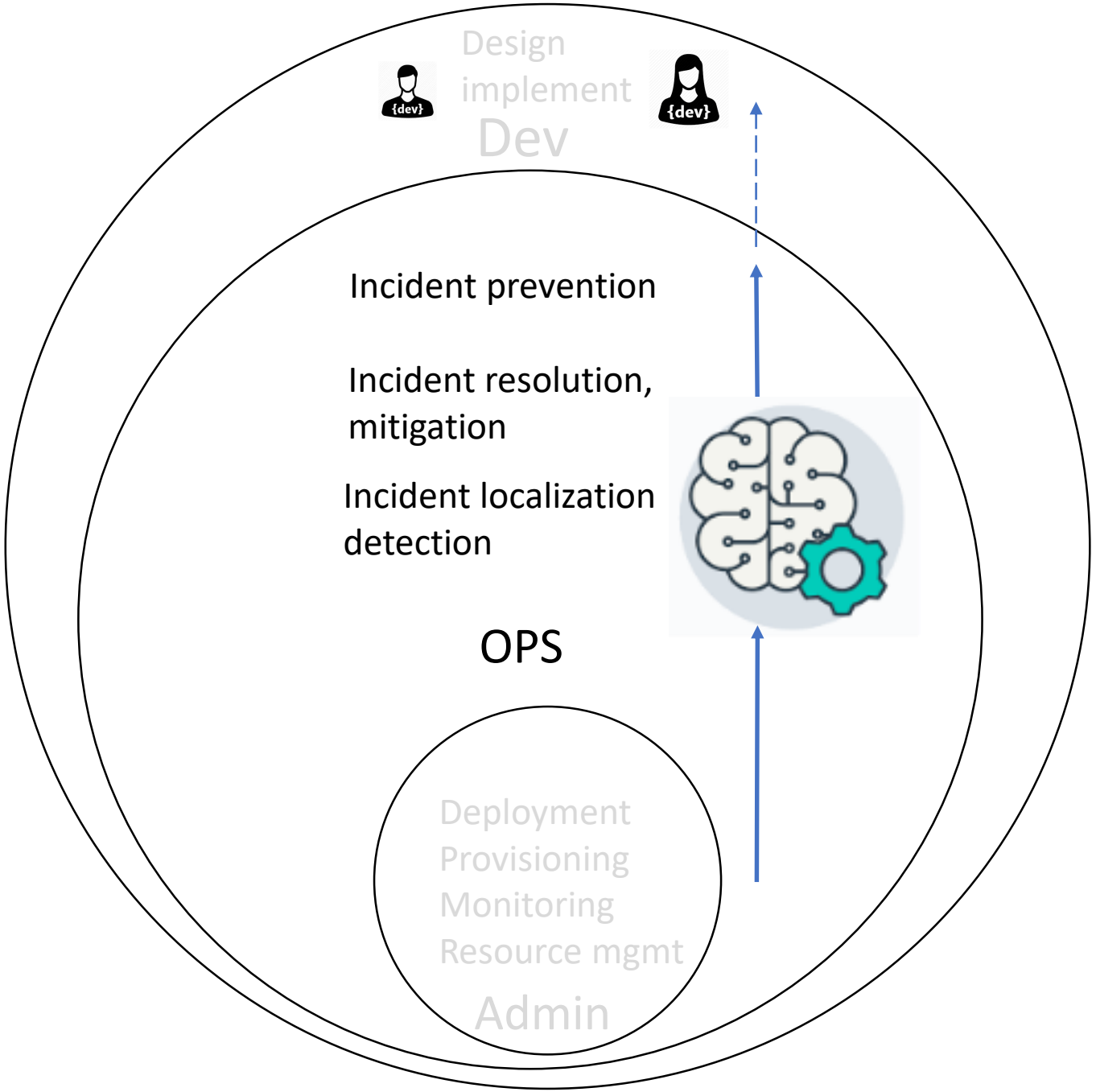
- Dependent services (ToRs) need to provide one extra nine to target service (VMs)

ToRs are not on critical path for VMs to achieve five-nines availability

Deepview Insight: VMs and their Storage Co-location

- For load balancing, VMs can mount VHDs from any storage cluster in the same region
- Some VMs have storage that are further away
- **Can longer network paths impact VM availability?**
- At Azure, 52% two-hop, 41% three-hop
- Compute daily VHD failure rates: r_0 (two-hop), r_1 (three-hop)
- Average over 3-months
- **Yes!** $(\bar{r_1} - \bar{r_0})/\bar{r_0} = 11.4\%$ **increase**

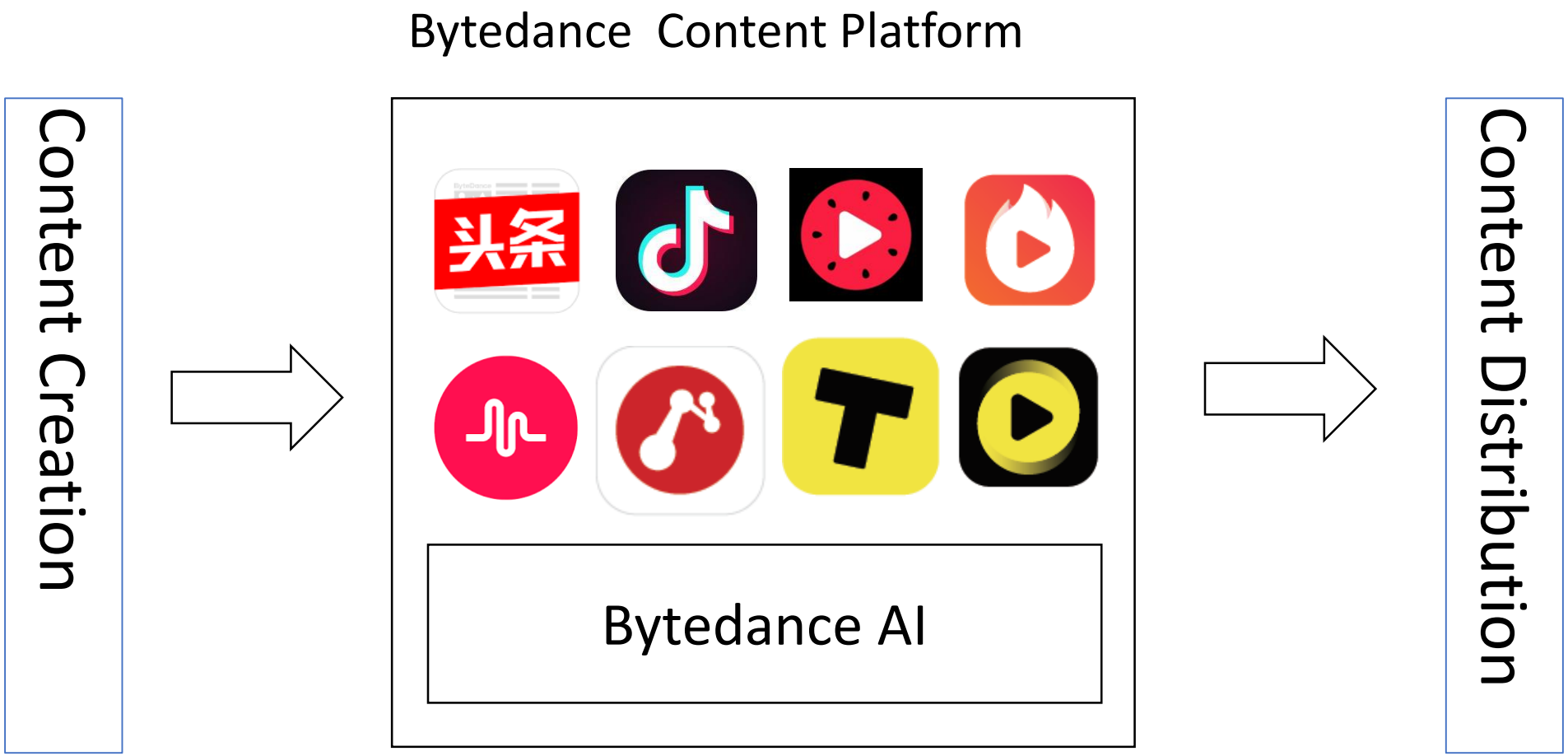
Some benefit to co-locate VM and their VHDs


Design
implement
Dev
Incident prevention
Incident resolution,
mitigation
Incident localization
detection
OPS
Deployment
Provisioning
Monitoring
Resource mgmt
Admin

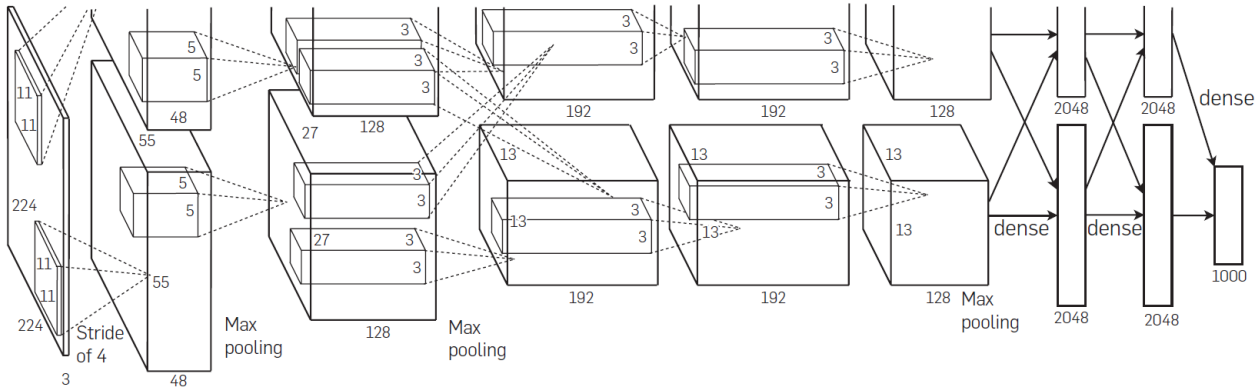
RDMA for ML Training Acceleration

-- A case where networking helps ML to scale

Background


Bytedance Content Platform
Content Creation
头条
Bytedance AI
Content Distribution

Content Understanding using DNN

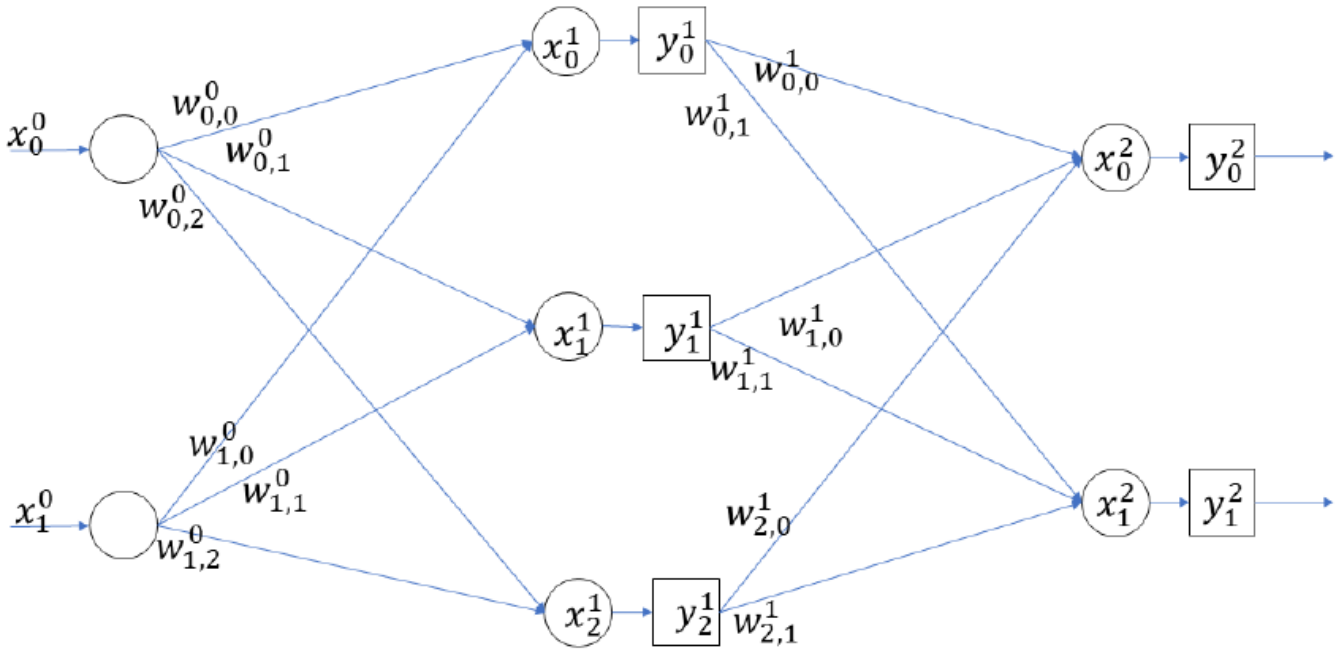


cat

AlexNet

DNN Training: BP

Forward



$$e = \frac{1}{2}\sum_{i=0,1}(y^2 - y_i^t)^2$$

Backward

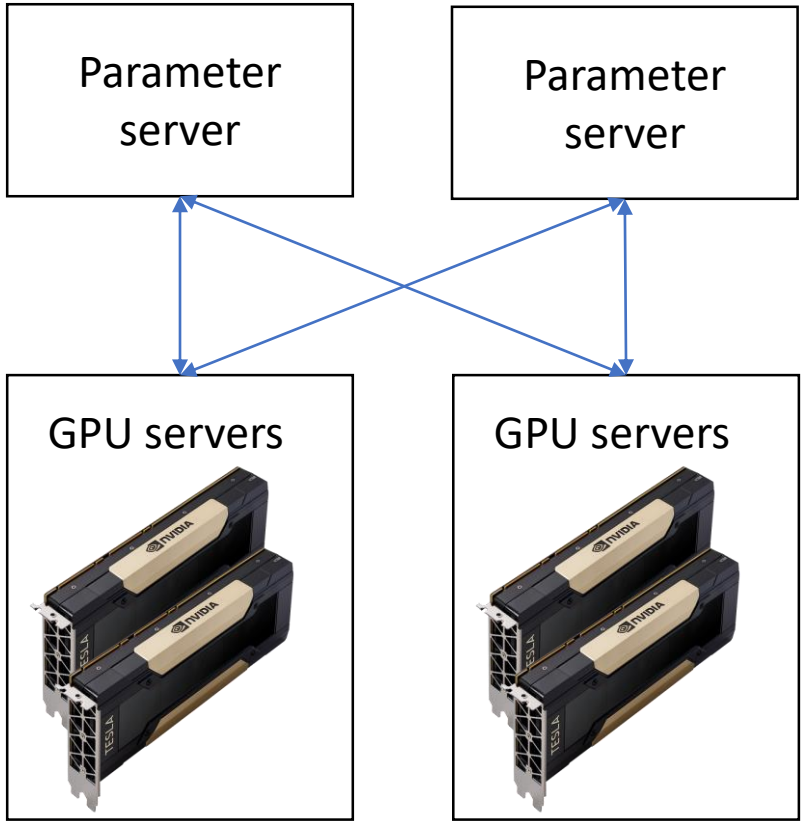
$$\begin{bmatrix} \frac{\partial e}{\partial w_{00}^0} & \frac{\partial e}{\partial w_{10}^0} \\ \frac{\partial e}{\partial w_{01}^0} & \frac{\partial e}{\partial w_{11}^0} \\ \frac{\partial e}{\partial w_{02}^0} & \frac{\partial e}{\partial w_{12}^0} \end{bmatrix} = \begin{bmatrix} \frac{\partial e}{\partial x_0^1} \\ \frac{\partial e}{\partial x_1^1} \\ \frac{\partial e}{\partial x_2^1} \end{bmatrix} \begin{bmatrix} x_0^0 & x_1^0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial e}{\partial w_{00}^1} & \frac{\partial e}{\partial w_{10}^1} & \frac{\partial e}{\partial w_{20}^1} \\ \frac{\partial e}{\partial w_{01}^1} & \frac{\partial e}{\partial w_{11}^1} & \frac{\partial e}{\partial w_{21}^1} \end{bmatrix} = \begin{bmatrix} \frac{\partial e}{\partial x_0^2} \\ \frac{\partial e}{\partial x_1^2} \end{bmatrix} \begin{bmatrix} y_0^1 & y_1^1 & y_2^1 \end{bmatrix}$$

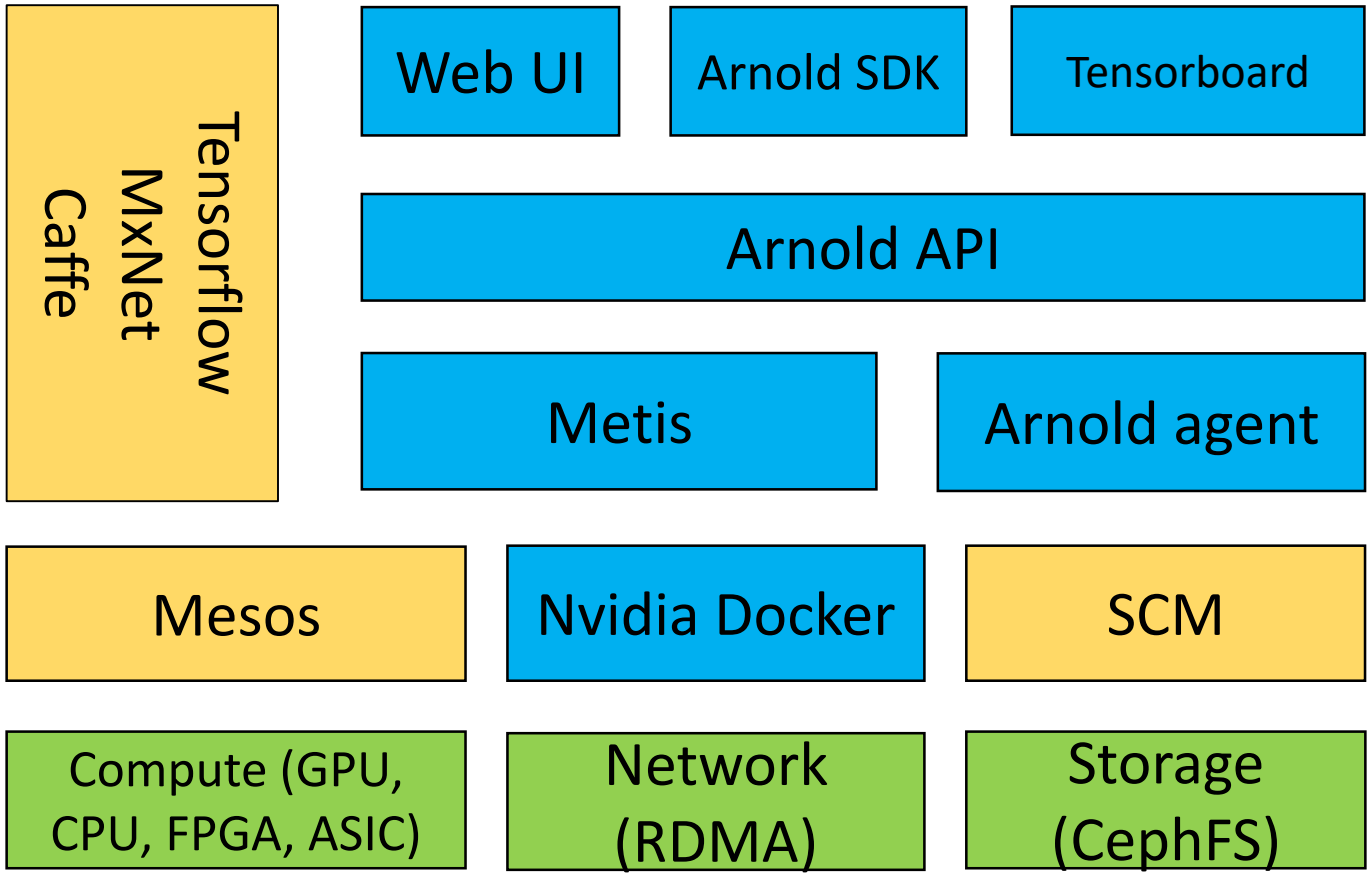
$$w = w - \eta \nabla w$$

Distributed Training Acceleration

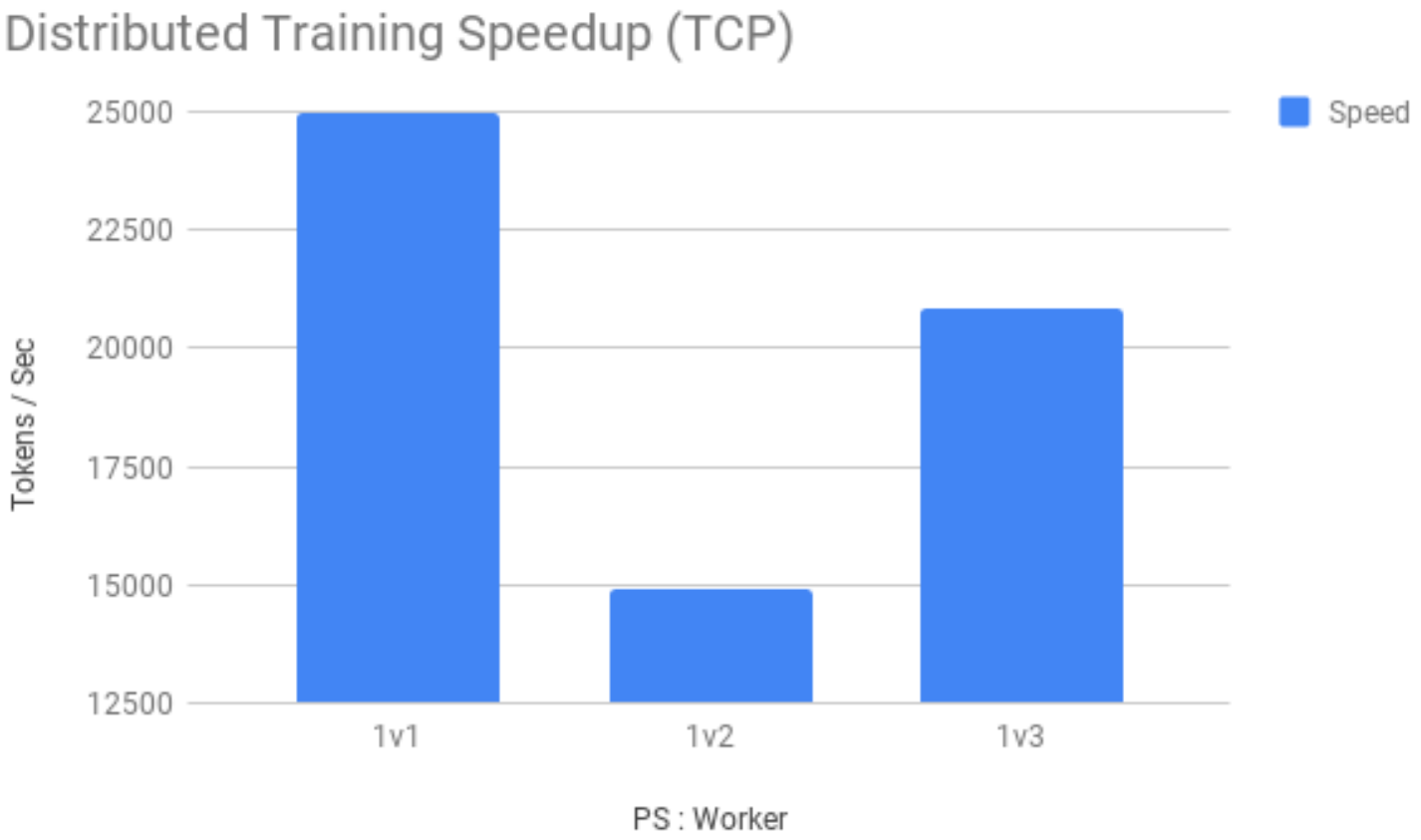
- GPU, with mini-batch
- Distributed training (data parallel)



Arnold Training System

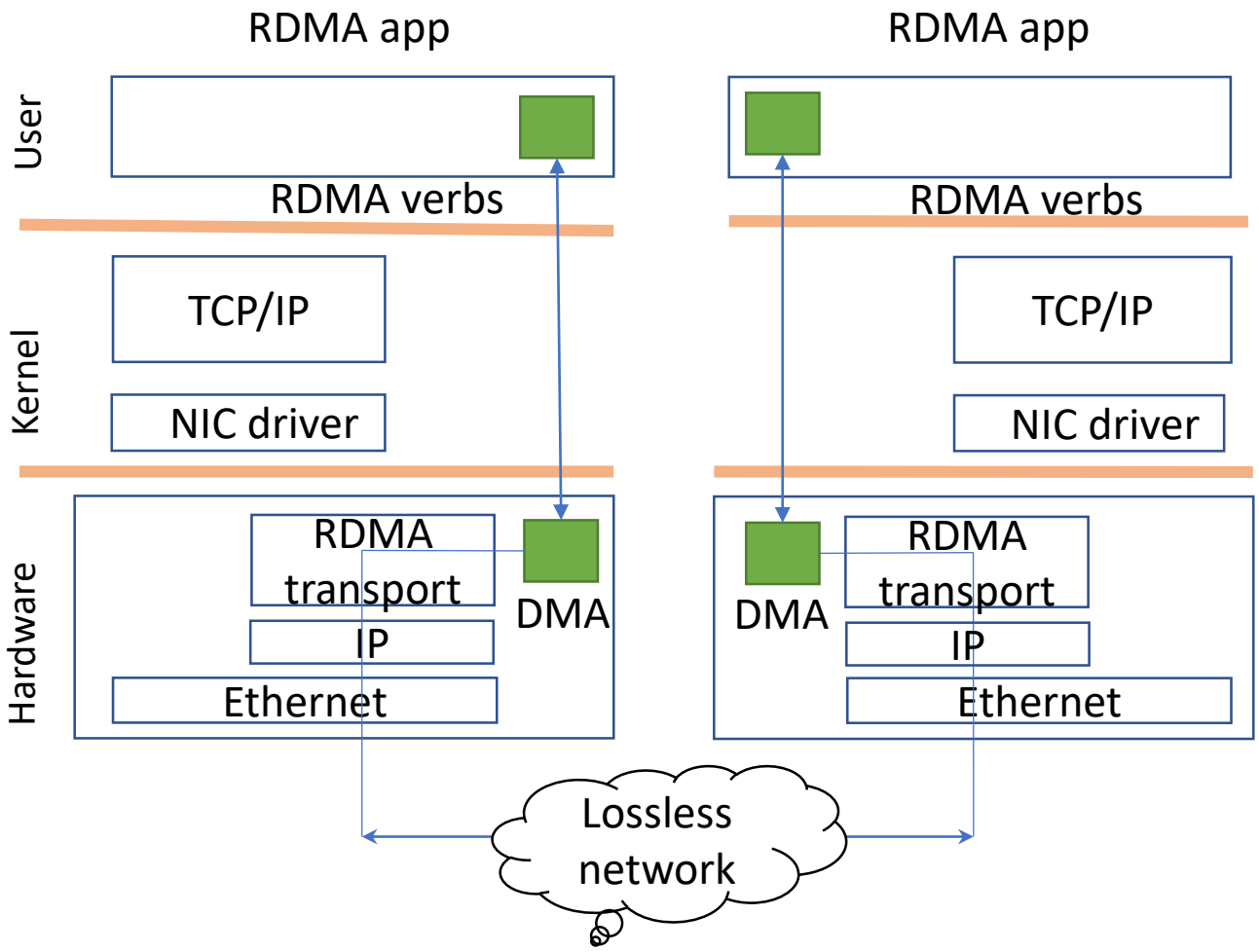


When Communication Becomes Bottleneck

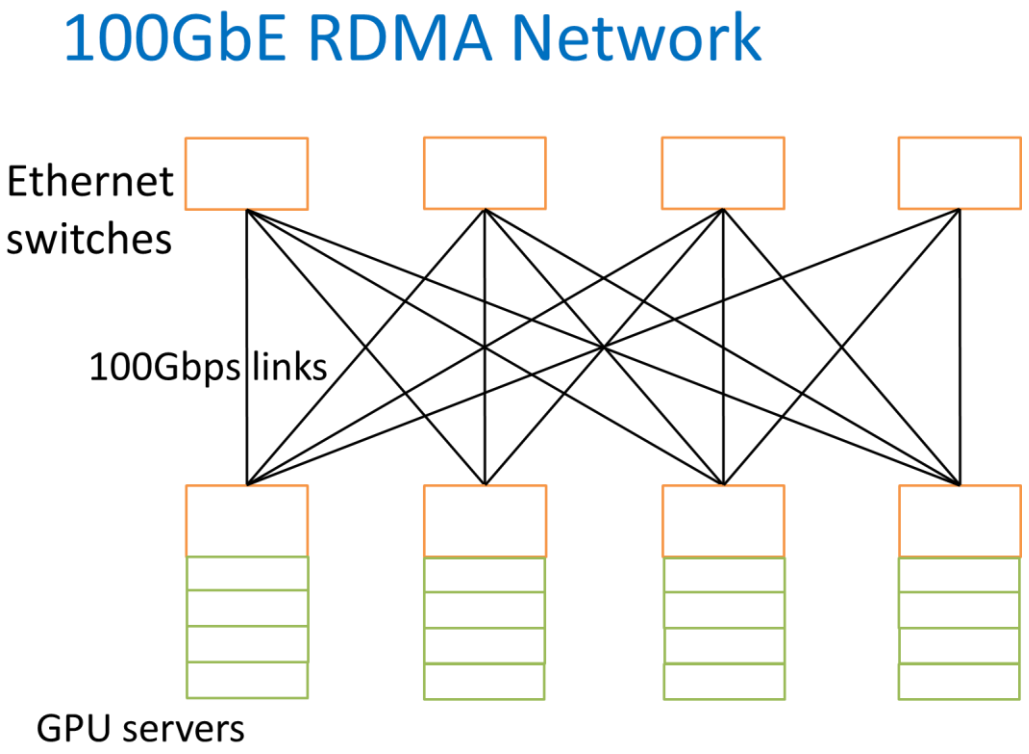


RDMA/RoCEv2 background

- RDMA addresses TCP's latency and CPU overhead problems
  - RDMA offloads the transport layer to the NIC
  - RDMA needs a lossless network
- RoCEv2: RDMA over commodity Ethernet
  - PFC for hop-by-hop flow control
  - DCQCN for connection-level congestion control [sigcomm15]
  - Many issues addressed [sigcomm16, conext17]



RDMA Cluster for Arnold Training



- 100Gbps throughput between any servers
- Micros-second e2e latency
- Minimal CPU overhead for packet processing

- Many models spend large amount of time on communication
  - x Poor TCP performance
  - x Low network bandwidth
- 100GbE RDMA network
  - ✓ Much higher bandwidth
  - ✓ Reduces communication time
  - ✓ Scales the cluster to thousands of GPU cards

RDMA Many-To-One

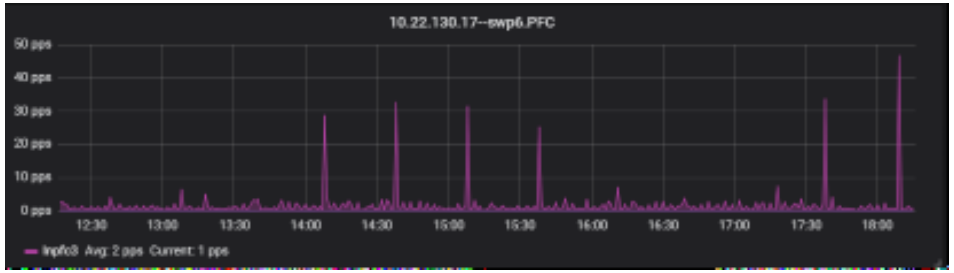
Throughput

switch

100GbE

Receiving server

PFC

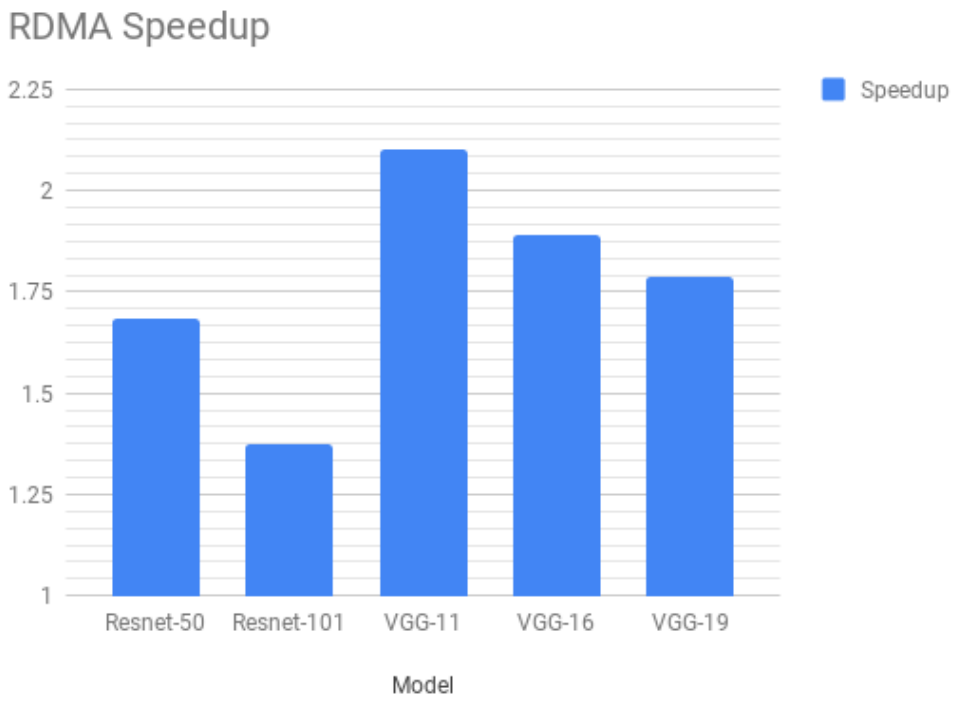


ECN

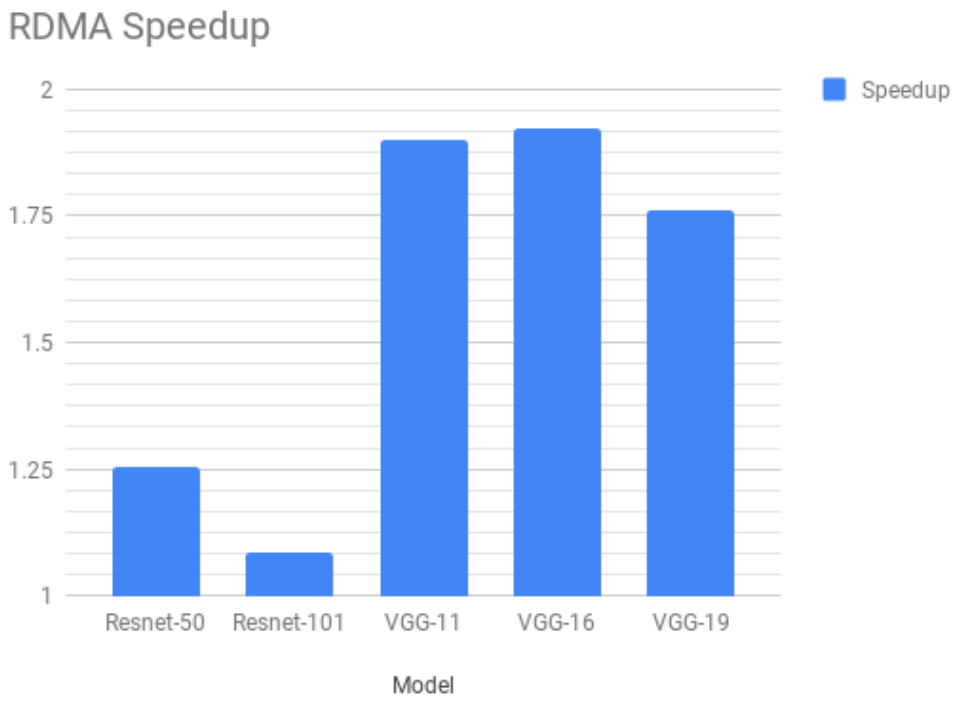
100GbE

Sending servers

RDMA for ML Training Acceleration (CNN)

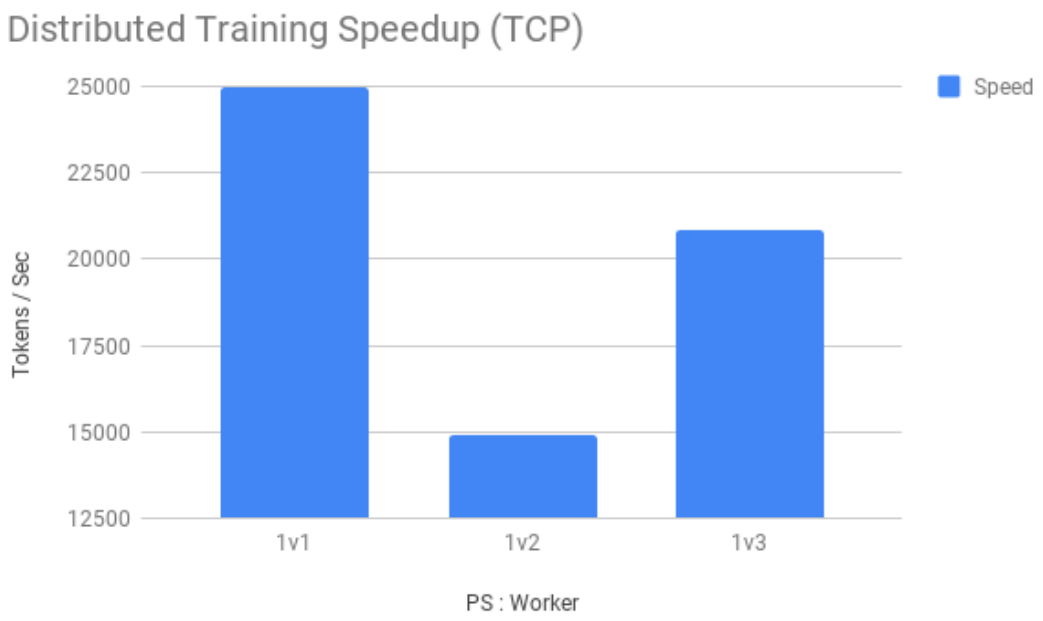


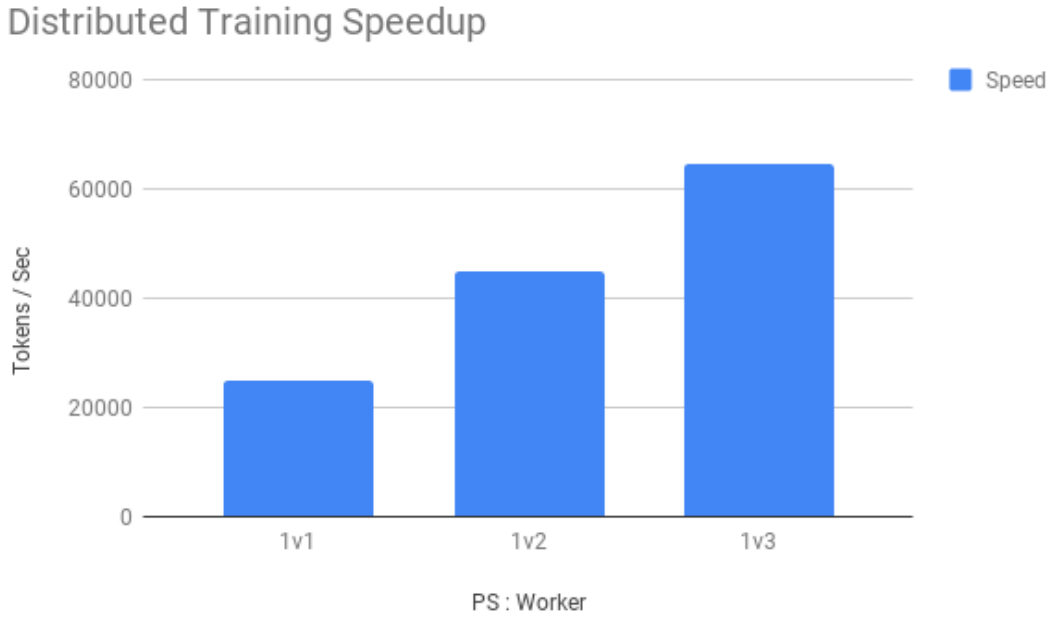
Batch size: 32



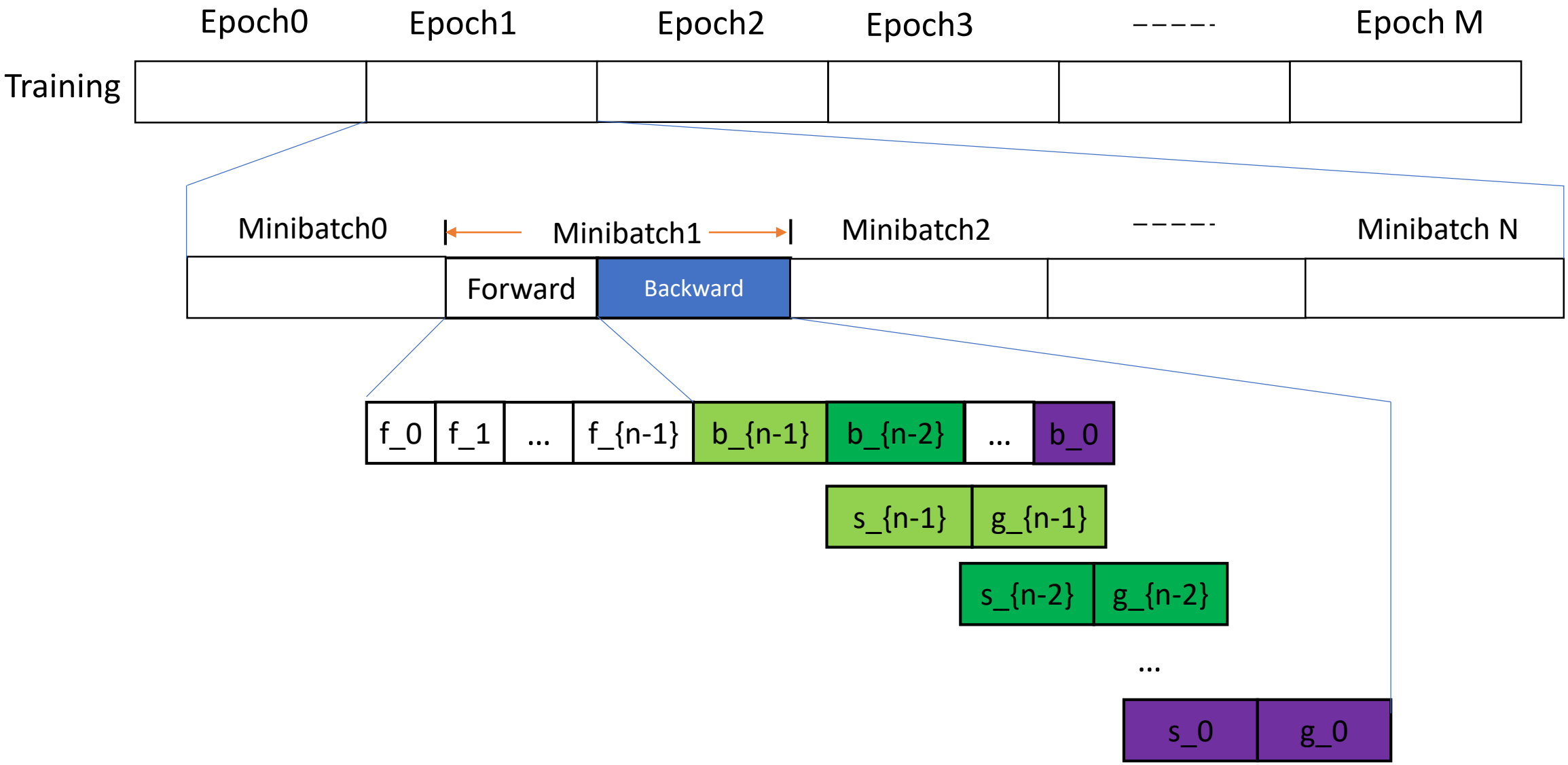
Batch size: 64

RDMA for ML Training Acceleration (RNN)





When RDMA Acceleration Helps



When RDMA Acceleration Helps

- Big models
  - ResNet50 (98MB), VGG19 (548MB)
- Communication/computation ratio is large
  - Layers with large parameter size
  - Small minibatch size
  - When TCP is slow

Summary

- ML will be a core part for building highly available systems
  - Deeper availability understanding
  - Automatic incident localization, mitigation, prevention
  - Intelligent system/network design
- System/networking for ML
  - Scalable ML systems
  - Hardware, systems, ML services integrated design

Acknowledgement

- Deepview (nsdi18): Qiao Zhang, Guo Yu, Yingnong Dang, Nick Swanson, Xinsheng Yang, Randolph Yao, Murali Chintalapati, Arvind Krishnamurthy, and Thomas Anderson
- ByteDance Networking Team
- Bytedance ML System Team

Q&A

ByteDance
字节跳动

Scan for More Info