Sibylla: To Retry or Not To Retry on Deep Learning Job Failure

Taeyoon Kim, Suyeon Jeong, Jongseop Lee
Soobee Lee, Myeongjae Jeon

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY
Shared multi-tenant GPU clusters today

Shared GPU cluster is commonly used to run DL workloads

Resource scheduling
Storage for data & model
Failure handling

Cluster Manager
Problem: Job failure of DL training

Unsuccessful job completion with job failures (resource waste)
Prior studies: failure root cause and impact analysis [1, 2]

How to deal with various failures to enhance the cluster resource utilization?

Two types of job failures

Deterministic failure (DT failure)
- Failure will repeat with retry on failure
  Ex) Syntax errors, API misuse, corrupted data

Non-deterministic failure (NDT failure)
- Failure is transient and can be overcome with retry on failure
  Ex) Network failures, MPI daemon errors
Existing approaches for failure handing

Fixed number of retries on failed jobs
(+) Increase job success rate (retrying NDT failures)
(−) Waste resources (retrying DT failures)

Termination of failed jobs
(+) Avoid worthless retry on DT failures
(−) Lower job success rate (not retrying NDT failures)
Sibylla: Predicting DT vs. NDT failure

**Goal** No retry on DT failure and retry on NDT failure

---

### Failure classifier in Sibylla

1. Highly accurate with various training error logs
2. Continuously updating without human intervention
Opportunity on predictive retry

Analysis using Microsoft Philly trace [1]

Resource inefficiency caused by DT failures

• 5–23% of jobs experience DT failures across job sizes
• 12–20% of GPU hours are wasted for retrying DT failures

Opportunity on predictive retry

Analysis using Microsoft Philly trace [1]

Resource inefficiency caused by DT failures
- 5–23% of jobs experience DT failures across job sizes
- 12–20% of GPU hours are wasted for retrying DT failures

Classifying failures by Sibylla can save *significant GPU hours wasted by frequent DT failures!*
Sibylla: Predicting DT vs. NDT failure

Goal No retry on DT failure and retry on NDT failure

Failure classifier in Sibylla

1. Highly accurate with various training error logs
   • RNN model-based classifier for determining DT/NDT

2. Continuously updating without human intervention
Data source: stderr/stdout streams

**Challenge** Unstructured and diverse log formats

```
in forward(self, x, hidden)
  ---> 17  x, hidden= self.lstm(x,hidden)
torch\nn\modules\module.py in _call_impl(self, *input, **kwargs)
  --> 727  result = self.forward(*input, **kwargs)
torch\nn\modules\rnn.py in forward(self, input, hx)
  --> 234  result = _impl(input, hx, ...)

TypeError: rnn_tanh() received an invalid combination of arguments - got (Tensor, Tensor, list, ...), but expected one of: * (Tensor, Tensor, Tensor, ...) didn't match because some of the arguments have invalid types: (Tensor, Tensor, !list!, ...)
```
Sibylla: Text preprocessing

Failed job log

Loading input is finished
Loading output is finished
Model loading success

Loading * is finished
Model loading success

[0.3, 0.2, ……, 0.6]
[0.6, 0.3, ……, 0.1]
Sibylla: Training phase

Failed job log (Labeled)

Log data correctly labeled by domain experts

RNN models (e.g., LSTM, GRU) to build the classifier
Sibylla: Predicting DT vs. NDT failure

**Goal** No retry on DT failure and retry on NDT failure

Failure classifier in Sibylla:

1. Highly accurate with various training error logs
2. Continuously updating without human intervention
   - Auto-labeling mechanism with classifier’s decision
Online logs auto-labeled for incremental model update
Auto-labeling based on an ensemble method
Tying all together

Kill or Retry

Cluster Manager

Failed job log

Sibylla

Training started
Can *Sibylla* improve cluster efficiency?

### Data collection
- 97 logs from a datacenter operator & 159 logs from Stack Overflow
- Augmented from 256 (97+159) to 4468 failure logs

### Training strategy
- 20% for initial training, then each 10% auto-labeled for updating classifier

### Comparison to Clustering, LSTM, GRU, and Oracle

---

Can *Sibylla* classify failure type well?

Sibylla outperforms other methods in classifying NDT failures

**Precision**

**Recall**
Can *Sibylla* classify failure type well?

*Sibylla* outperforms other methods in classifying NDT failures.

**Precision**

<table>
<thead>
<tr>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td>LSTM</td>
<td>GRU</td>
<td>Sibylla</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

97.36%

**Recall**

<table>
<thead>
<tr>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td>LSTM</td>
<td>GRU</td>
<td>Sibylla</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

98.66%

**Comparison Table**

<table>
<thead>
<tr>
<th></th>
<th>Sibylla</th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>97.36</td>
<td>98.66</td>
</tr>
<tr>
<td>Recall</td>
<td>98.66</td>
<td>98.66</td>
</tr>
</tbody>
</table>
Can *Sibylla* improve cluster efficiency?

Trace-driven simulation

- Job scheduling trace from Microsoft Philly
- Job execution simulator from Tiresias [1]

Three job scheduling policies

- Smallest Job First (SJF), 2D-LAS (DLAS), 2D-Gittins index (GITTINS)

Cluster specification

- 200 nodes, each 8 GPUs, 256GB of host memory, and 64 CPU cores

---

Can *Sibylla* improve cluster efficiency?

Comparison to

- Oracle: 100% correct predictions
- Full Retry: Retrying jobs w/o prediction (same as *Philly*)
Can *Sibylla* improve cluster efficiency?

Job completion time with Sibylla

- Improves 15.4% for SJF, 6.5% for DLAS and GITTINS than Full Retry
- Worsens only 1.0% compared with 100% correct prediction

---

**Can Sibylla improve cluster efficiency?**

**Job completion time with Sibylla**

- Improves 15.4% for SJF, 6.5% for DLAS and GITTINS than Full Retry
- Worsens only 1.0% compared with 100% correct prediction
Can *Sibylla* maintain job success rate?

Success rate on predictive retry
- Misprediction on failed job leads to lower job success rate

Compared to *Full Retry*
- *Full Retry* has highest job success rate
- *Sibylla* is lower the job success rate by only 0.06% from 75.04%
Conclusion

Job failure classifier

- *Sibylla*, predicting DT and NDT to help cluster kill DT and retry NDT

Performance of *Sibylla*

- *Sibylla* achieves consistently high performance on classifying failures
- Predictive retry with *Sibylla* can improve cluster efficiency