# ByteGAP: A Non-continuous Distributed Graph Computing System using Persistent Memory

Miaomiao Cheng , Jiujian Chen, ***Cheng Zhao***, Cheng Chen, Yongmin Hu, Xiaoliang Cong, Liang Qin, Hexiang Lin, Rong Hua Li, Guoren Wang, Shuai Zhang and Lei Zhang

***Douyin Vision Co., Ltd.*** & Beijing Institute of Technology

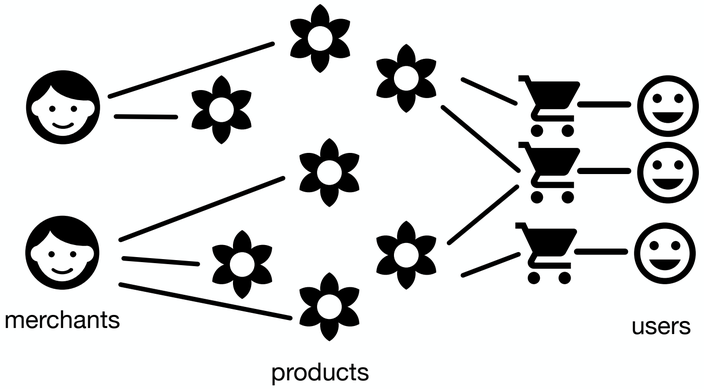
ByteDance 字节跳动

# Backgrounds

Risk Control Graphs

Social Networks

E-commercial Graphs

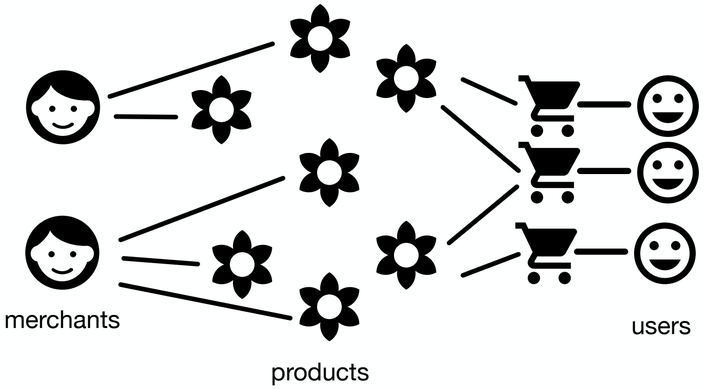


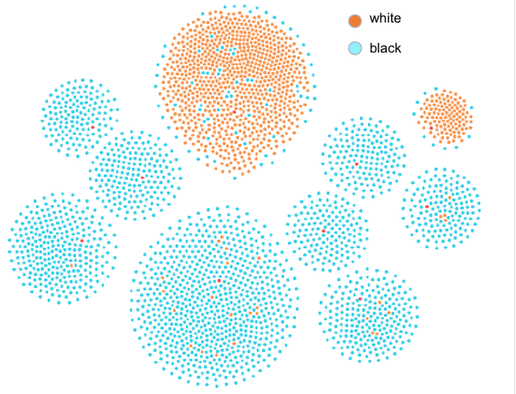
# Backgrounds

Risk Control Graphs

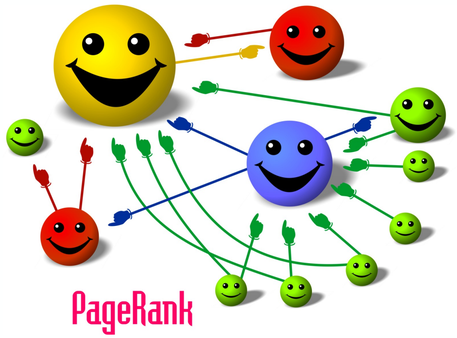
Social Networks

E-commercial Graphs

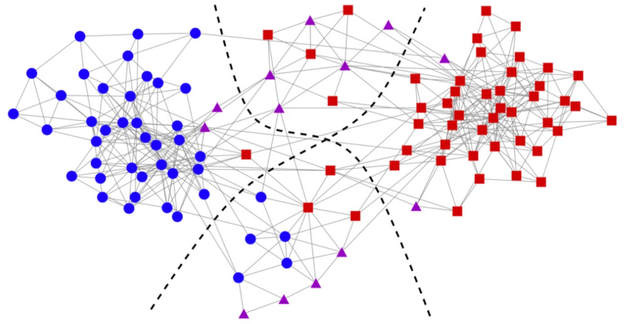



Fraud Detective



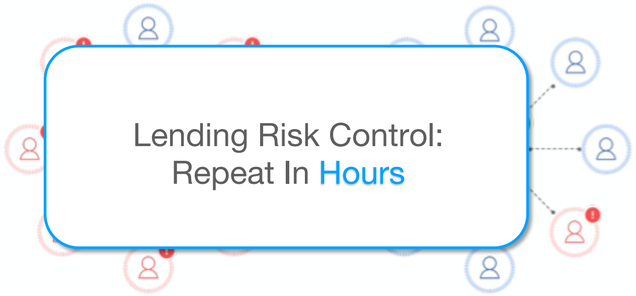

Centrality Rank



Vertex Classification/Clustering

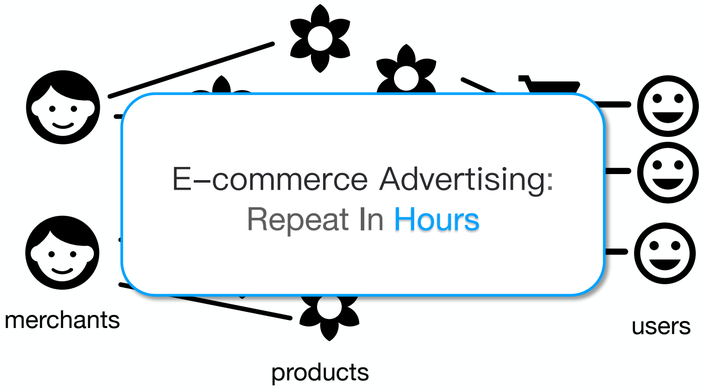
# Motivations

Graph Computation Tasks Run Periodically in ByteDance

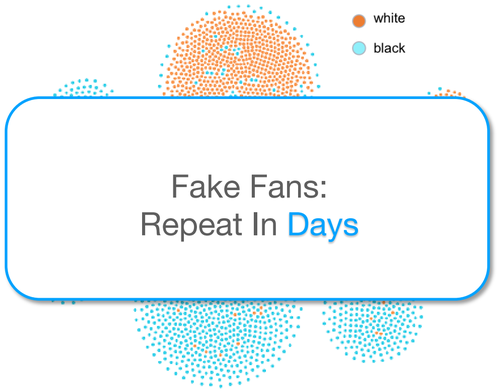


Lending Risk Control:
Repeat In Hours

Social Recommendation:
Repeat In Days/Weeks




E–commerce Advertising:
Repeat In Hours




Fake Fans:
Repeat In Days

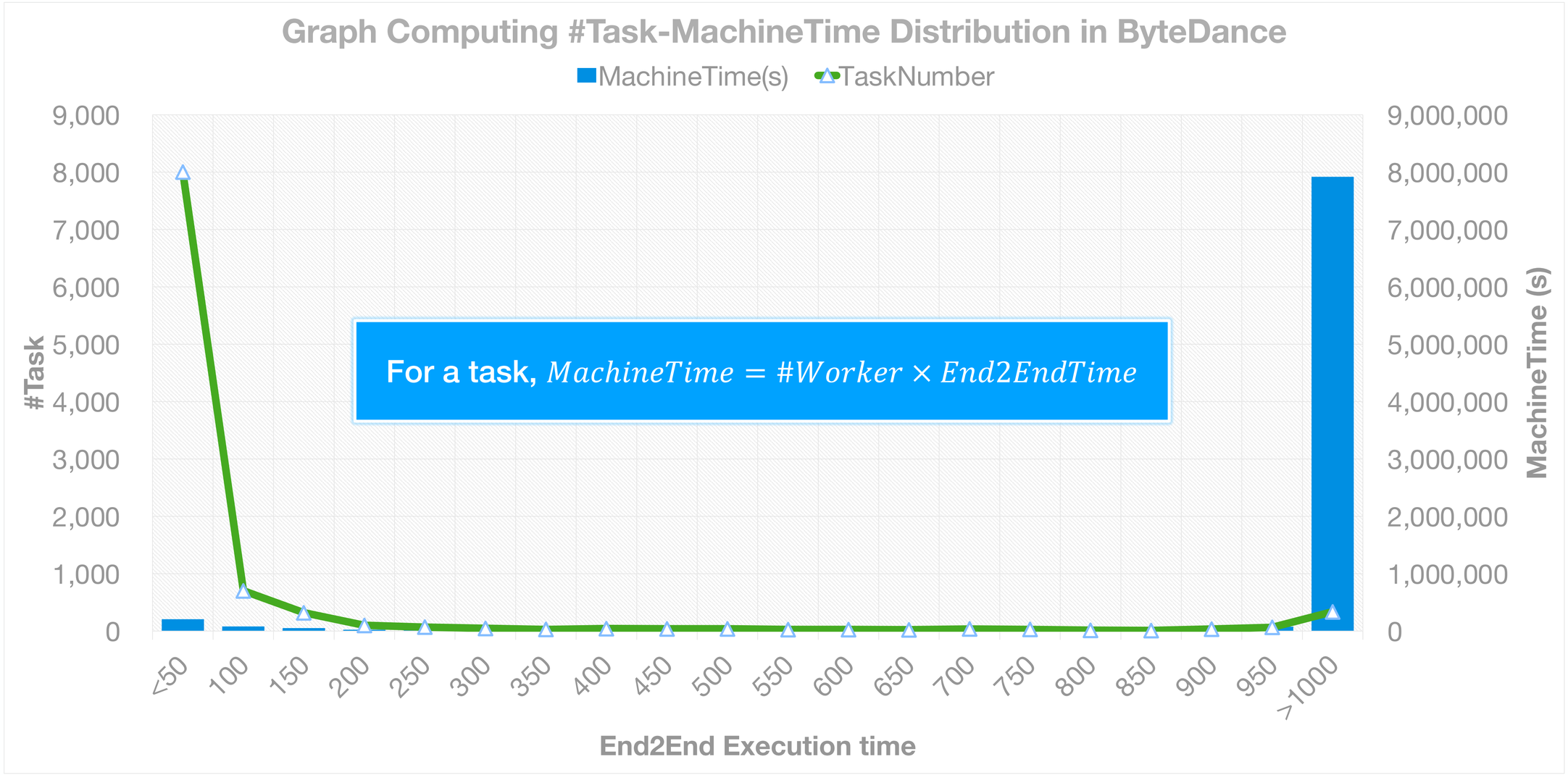

Hotspot Mining:
Repeat In Minutes

Centrality Rank

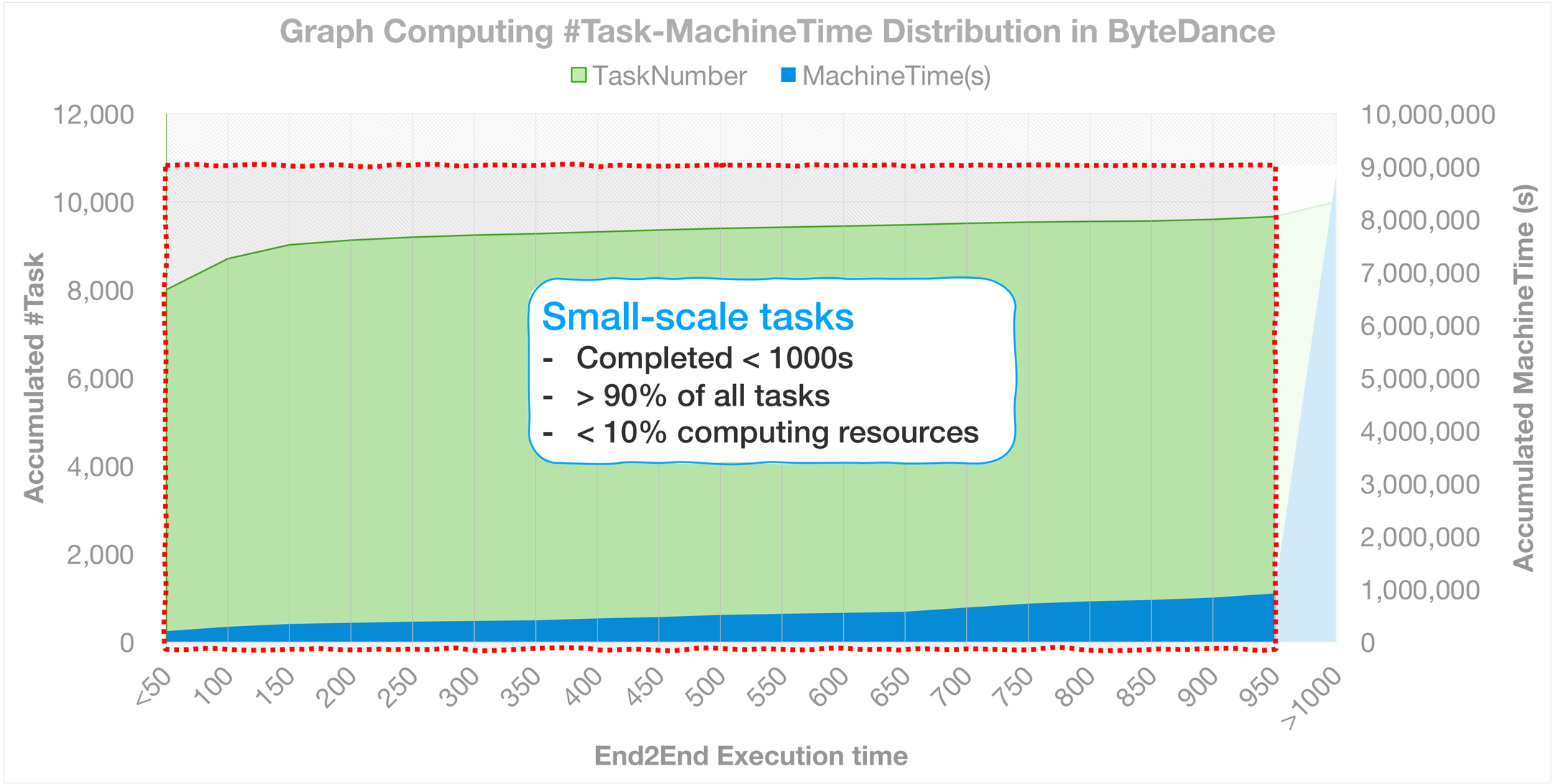
Product Clustering:
Repeat In Days/Weeks

Recommendation System

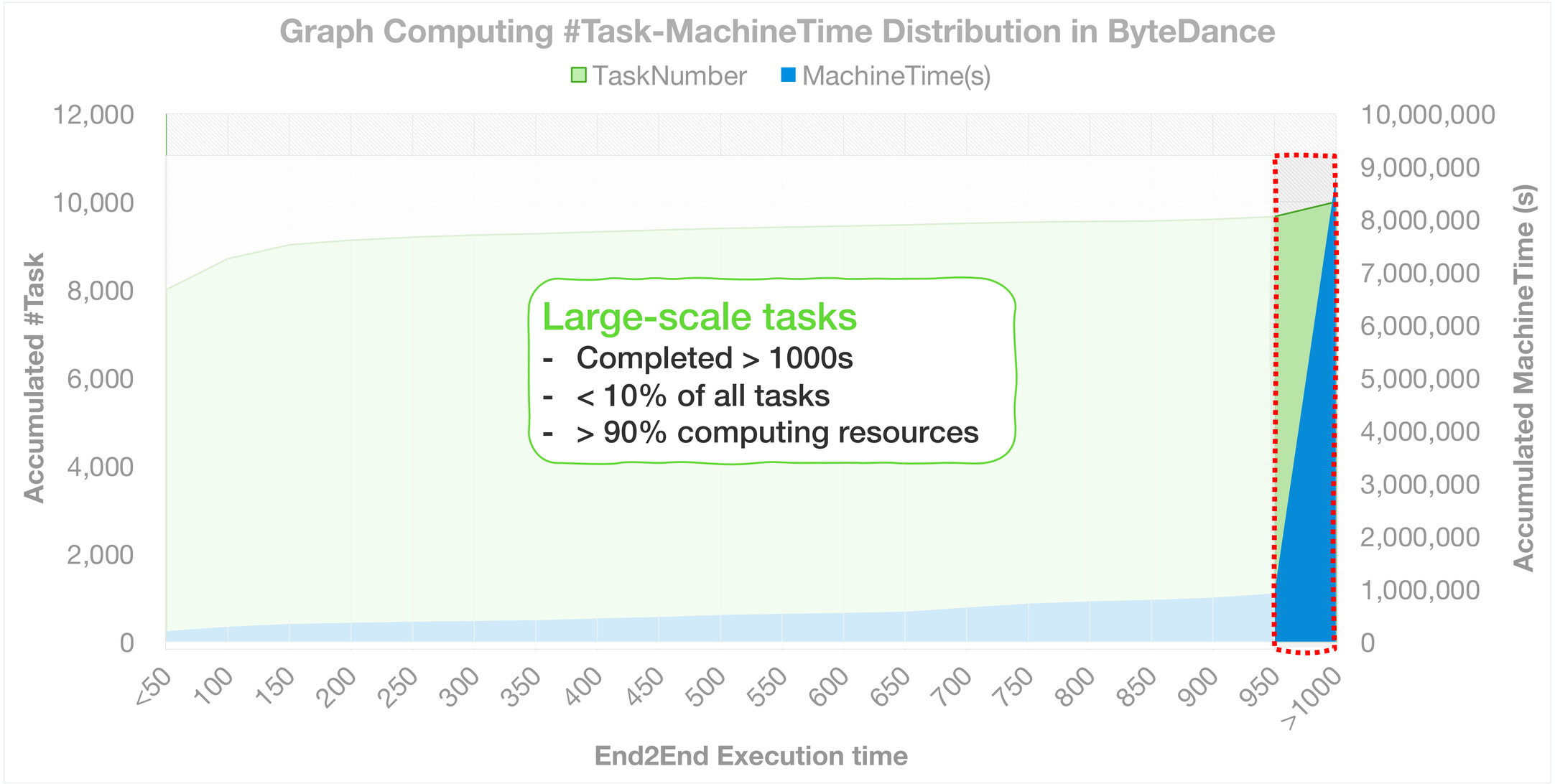
# Motivations



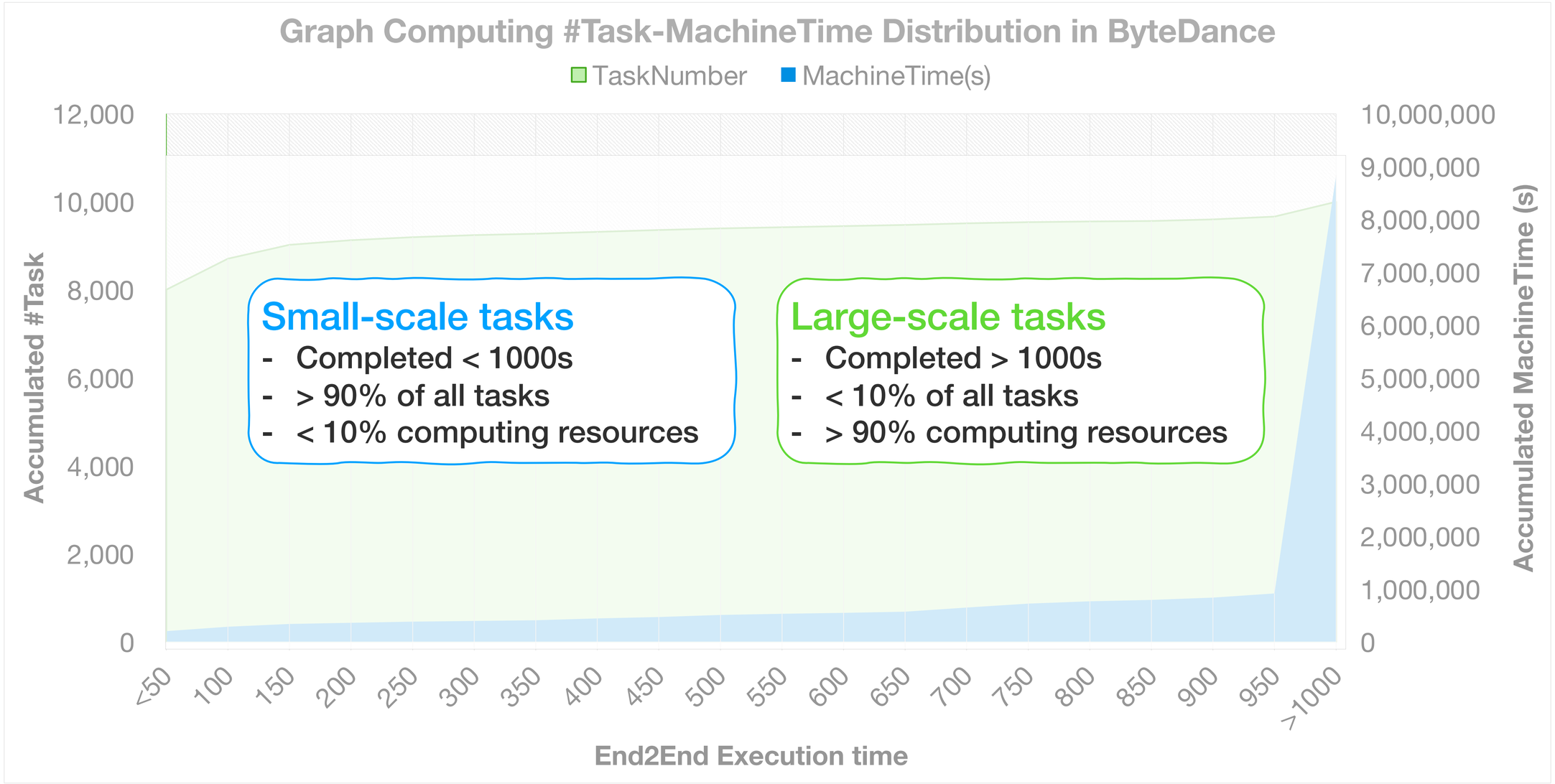
# Motivations



# Motivations

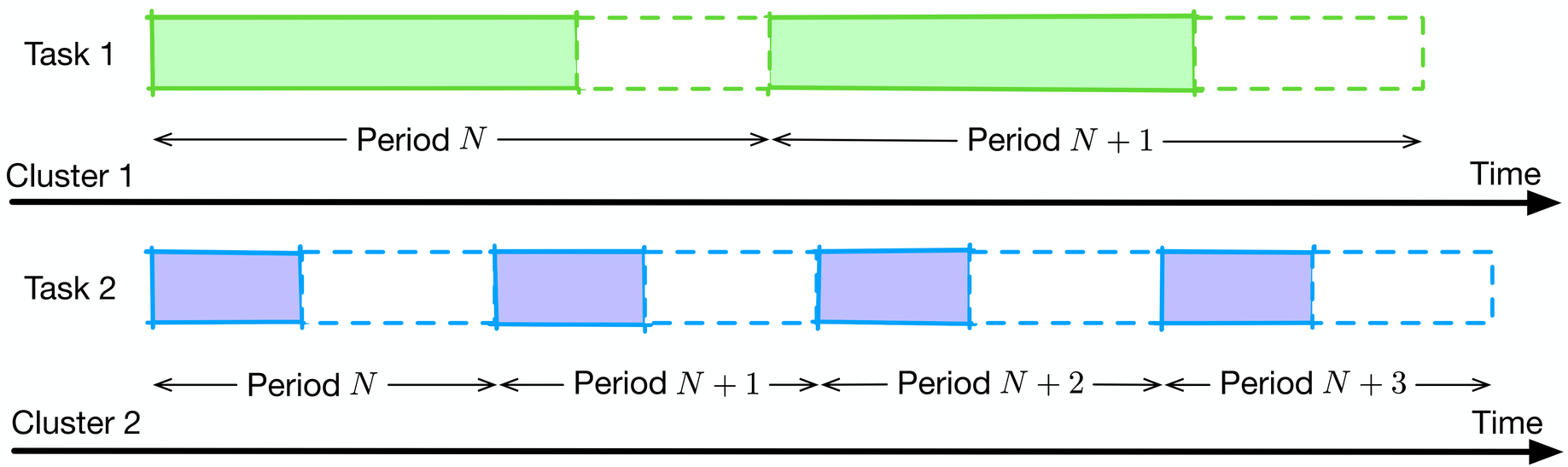


# Motivations



# Problems

Ideal…

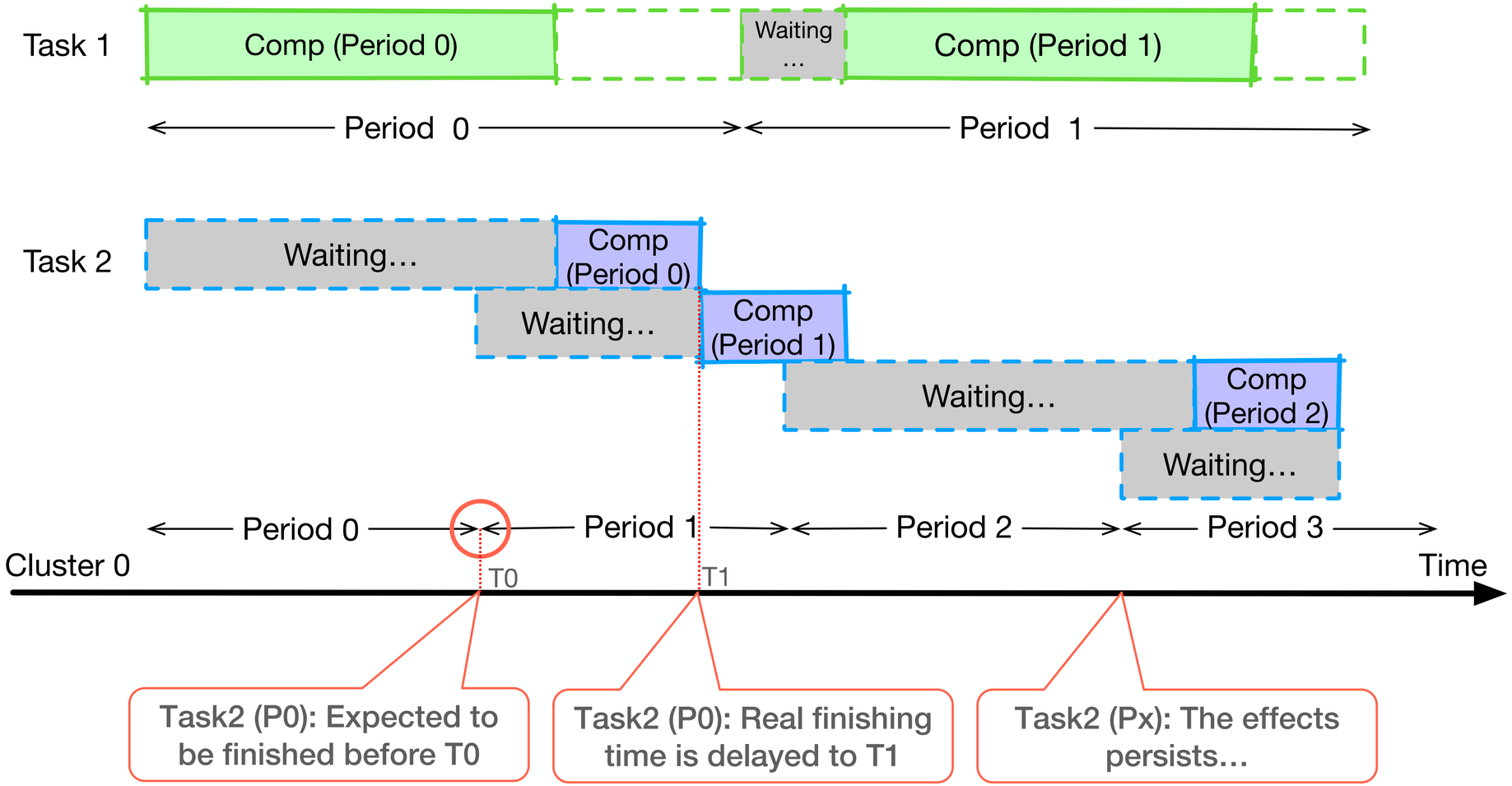



In real deal…

**Computing resources are scarce.**
**A cluster needs to handle multiple tasks.**

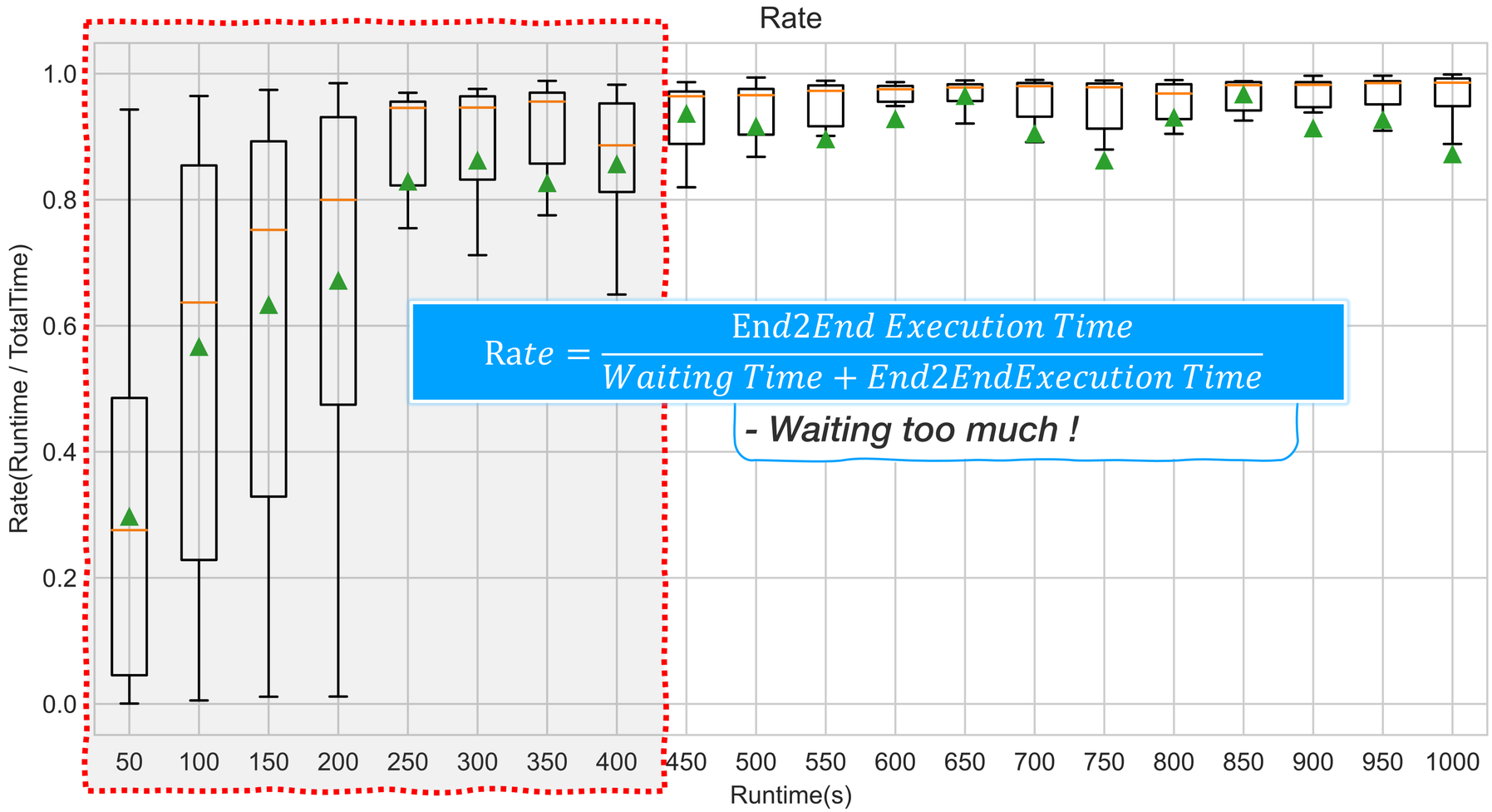
# Problems

In real deal…

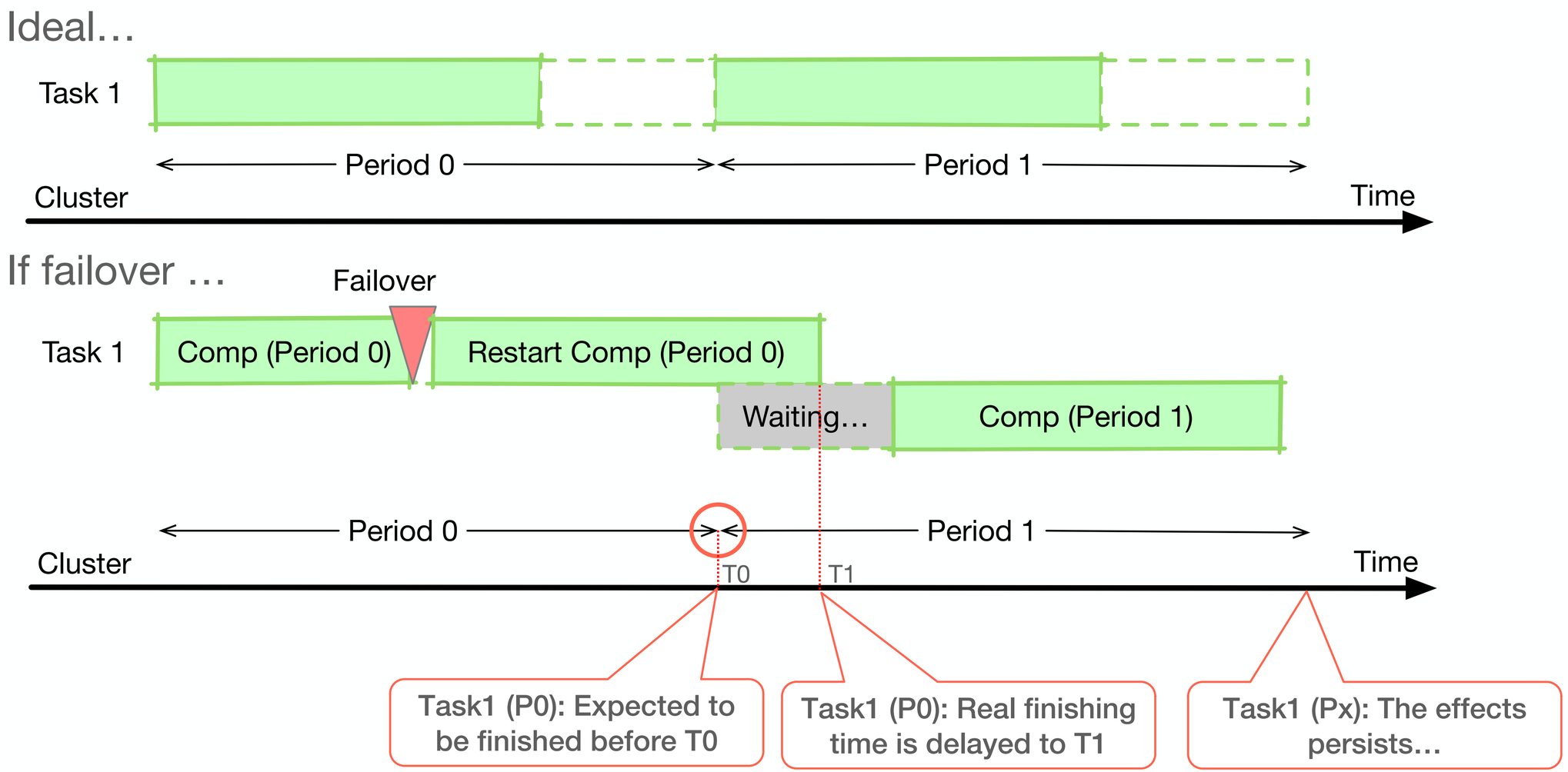


# Problems

In real deal…



# Problems



# Problems

*ByteDance's Graph Data*

- *|V|: 10 Billions+*
- *|E|: 1 Trillions+*

*Graph Computing Systems*

- *Distributed*
- *In-Memory*

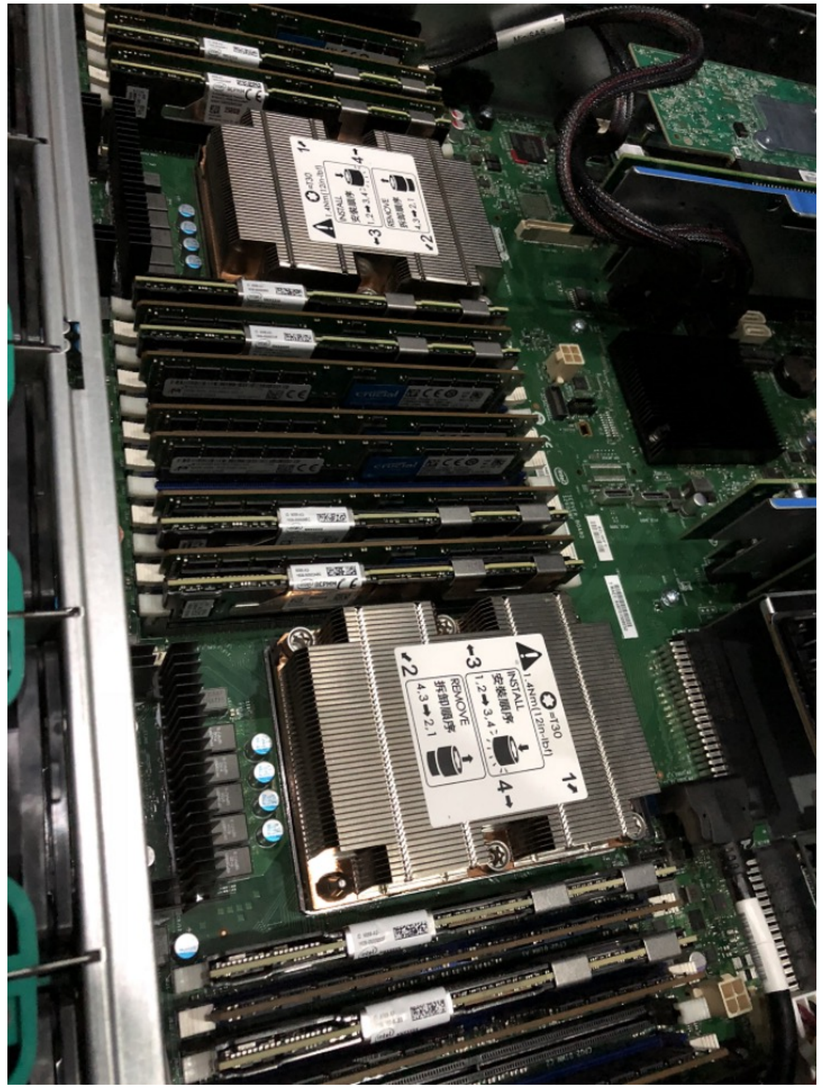
*Naïve Checkpointing*

- *10s TB~ 100s TB*
- *Ignore Iterativeness*

*Out-of-core Graph Computing Systems*

- *How to Checkpoint & Recover?*

# Problems



Intel® Optane™ Persistent Memory (PMEM)

## Capacity

**DRAM:** 4GB ~ 128GB

**PMEM:** 128GB ~ 512GB

*Complex Data Access Types*

*Vertex States* *Edge Data*

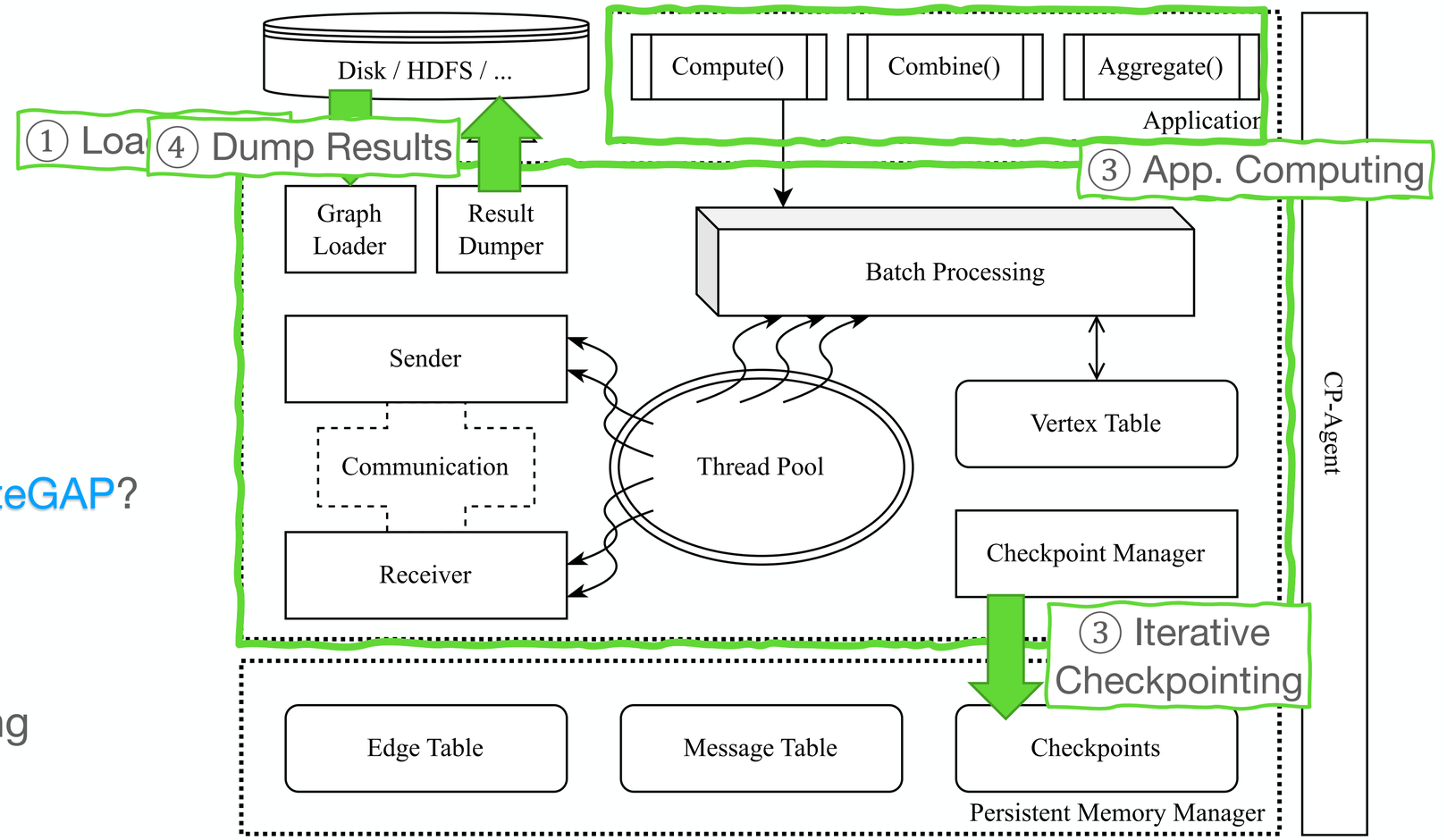
*Message Data*

# ByteGAP Overview: Examples

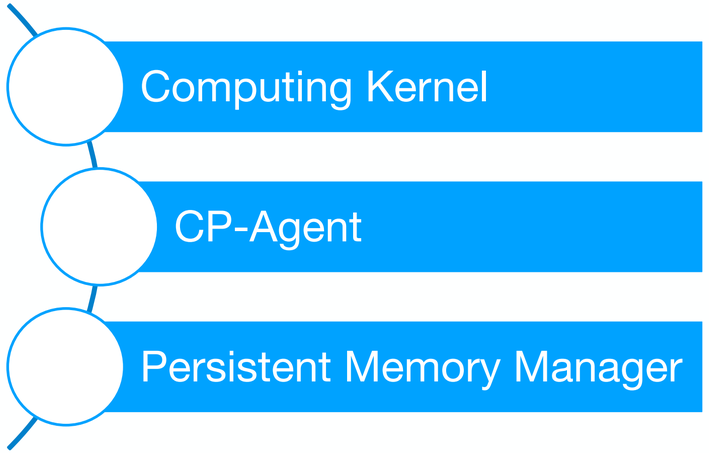
- Computing Kernel
- CP-Agent
- Persistent Memory Manager

Example: How to run a task in ByteGAP?

- Load Graph
- Initiate
- Repeat: Iteration & Checkpointing
- Dump Results



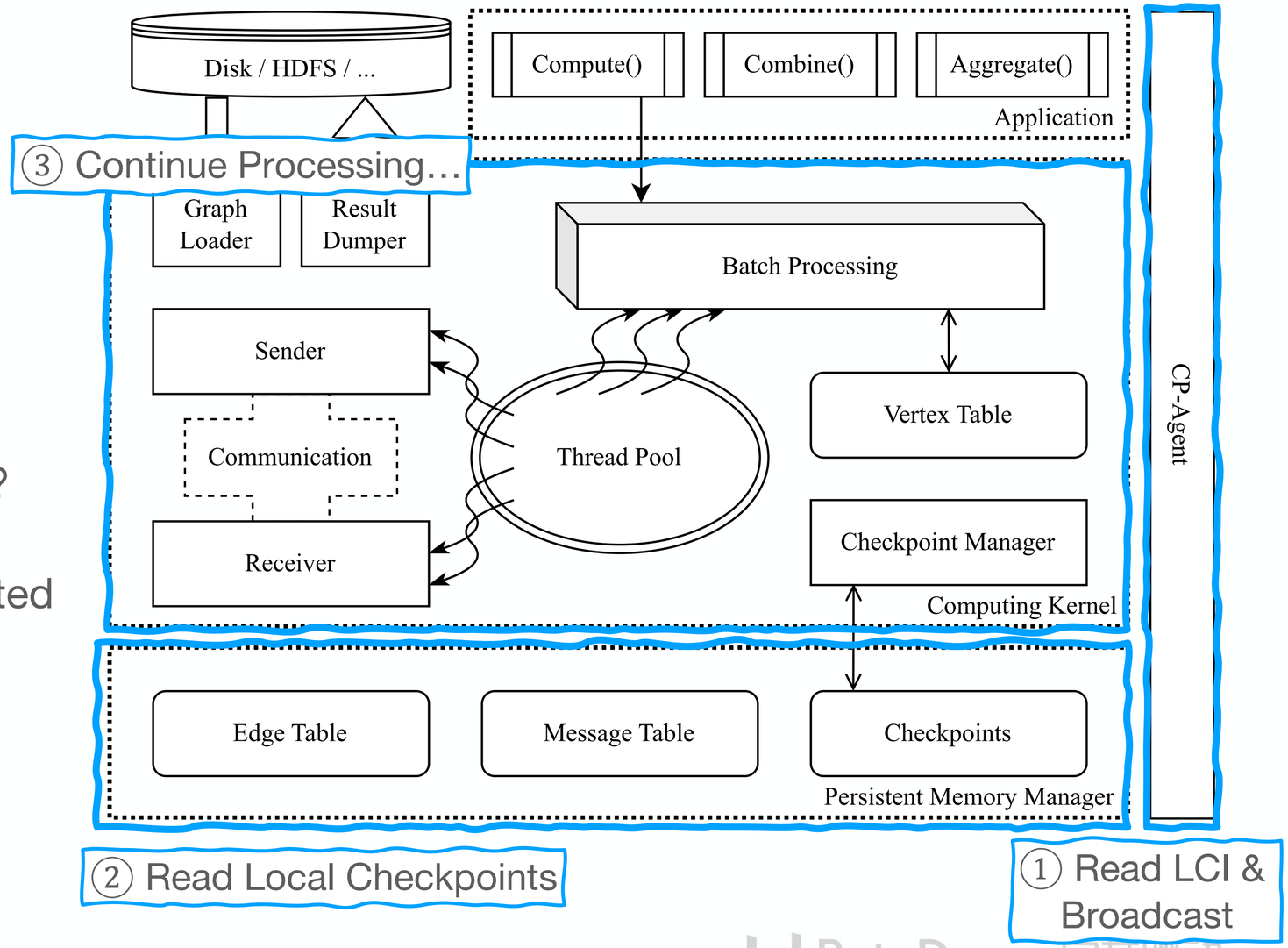
# ByteGAP Overview: Examples



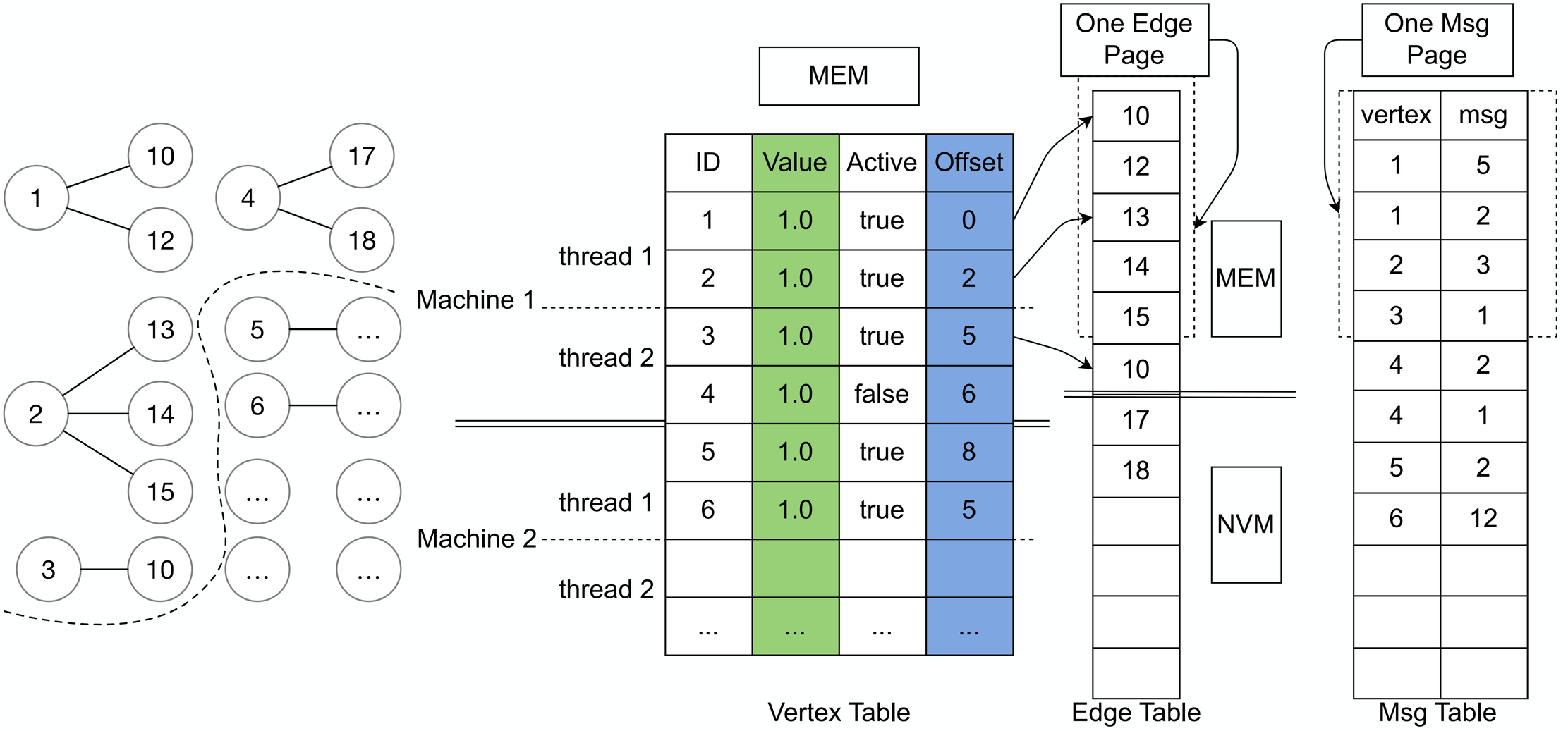

Example: How to run a task in ByteGAP?

Example: How to recover a failed/interrupted task in ByteGAP?

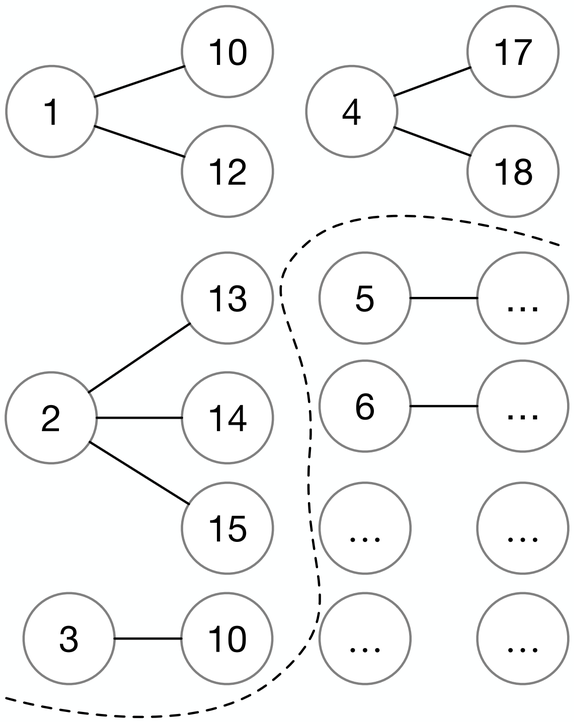
- Read Last Checkpoint ID
- Read Local Checkpoints
- Continue…

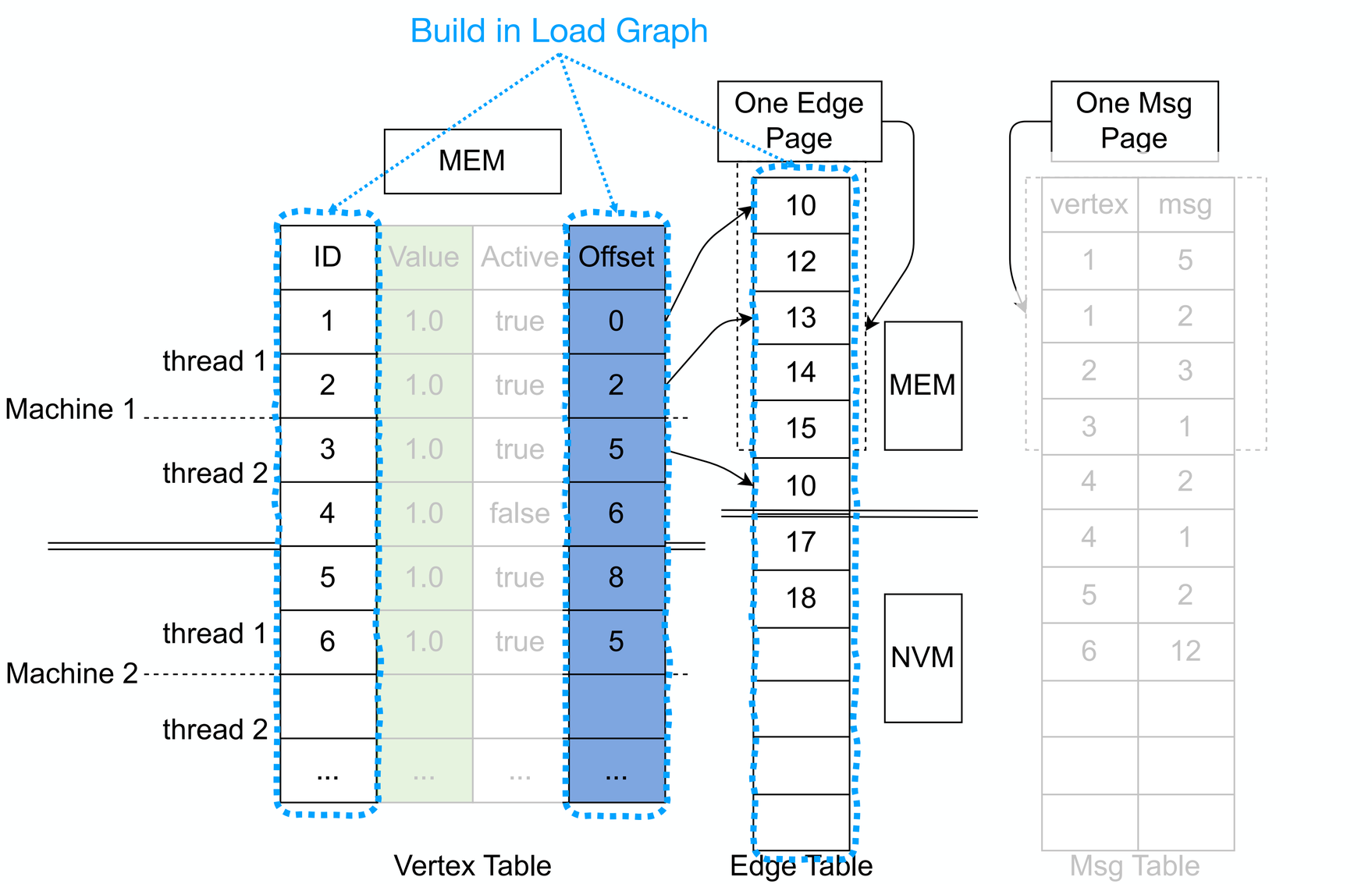


# Data Layout

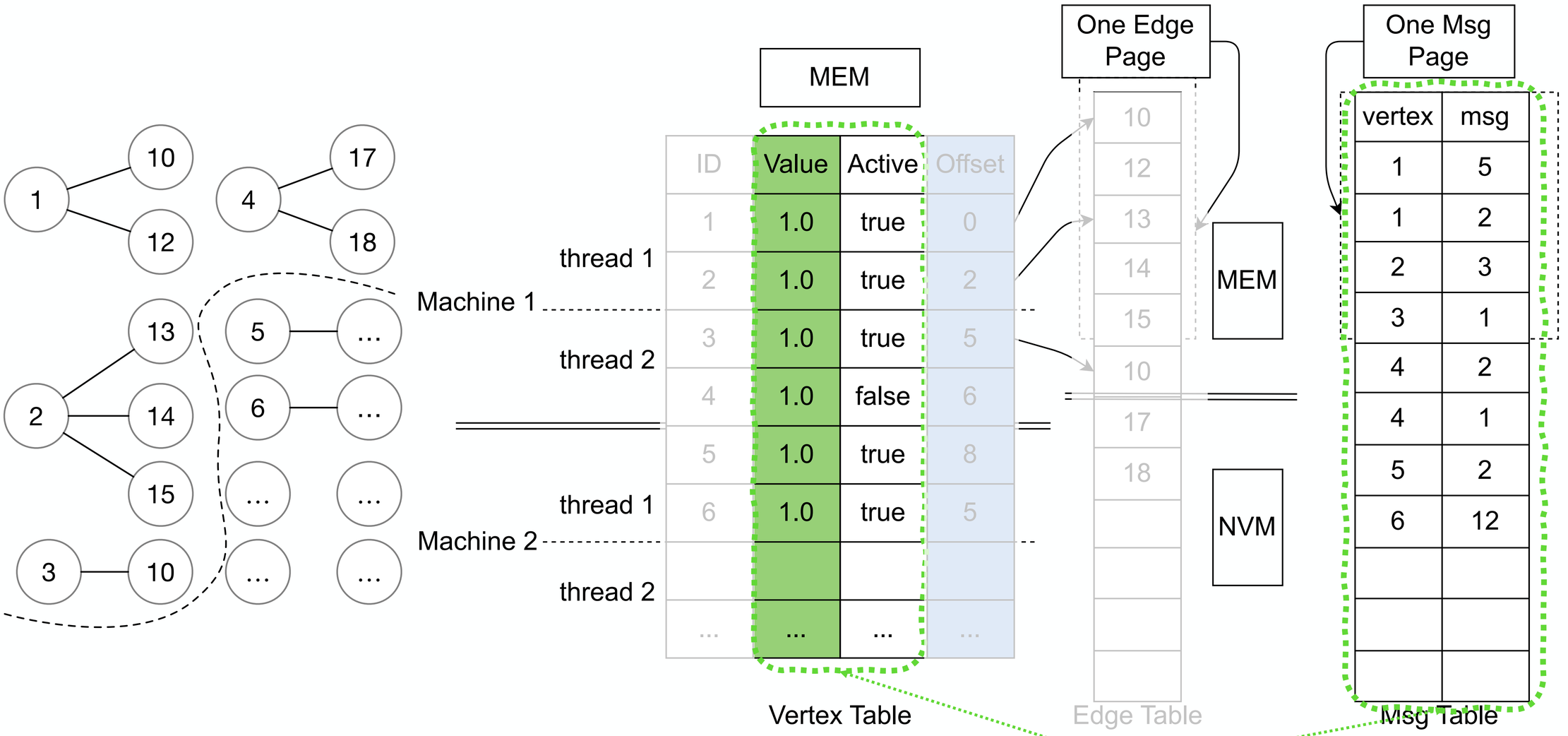


# Data Layout

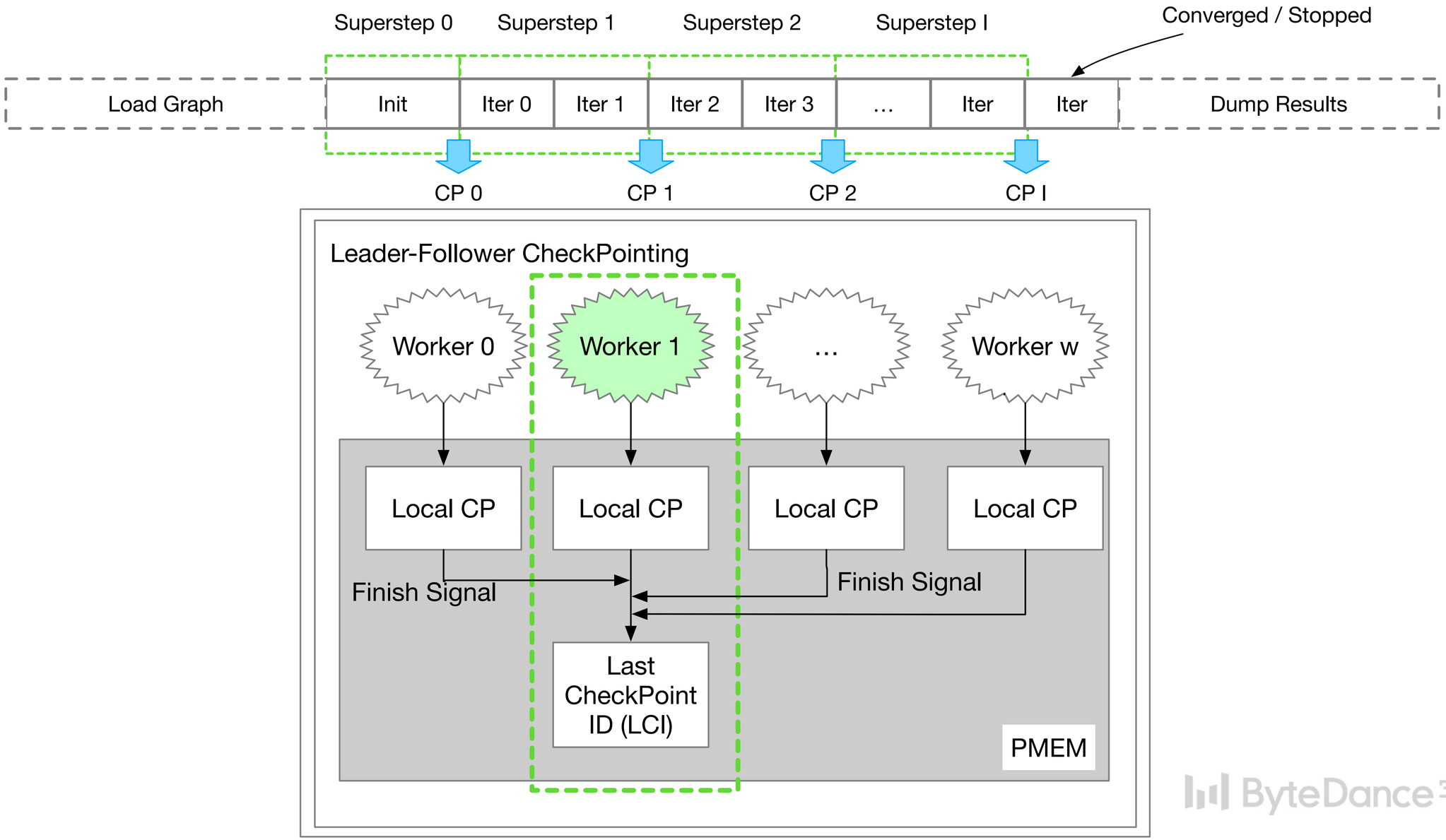


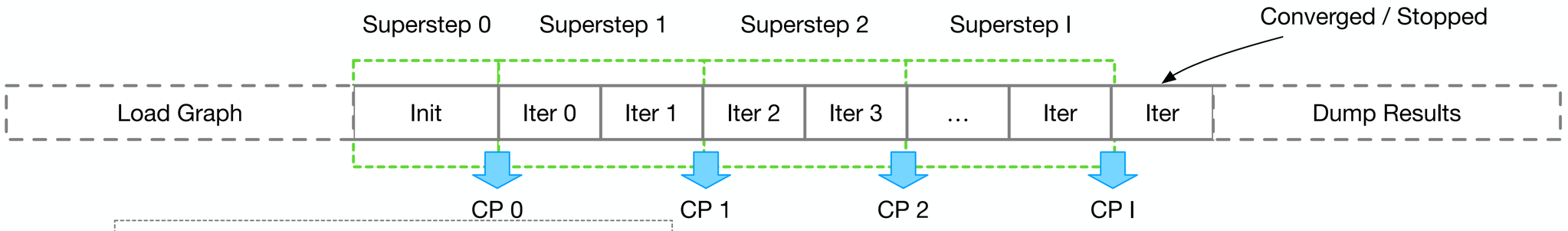
# Data Layout

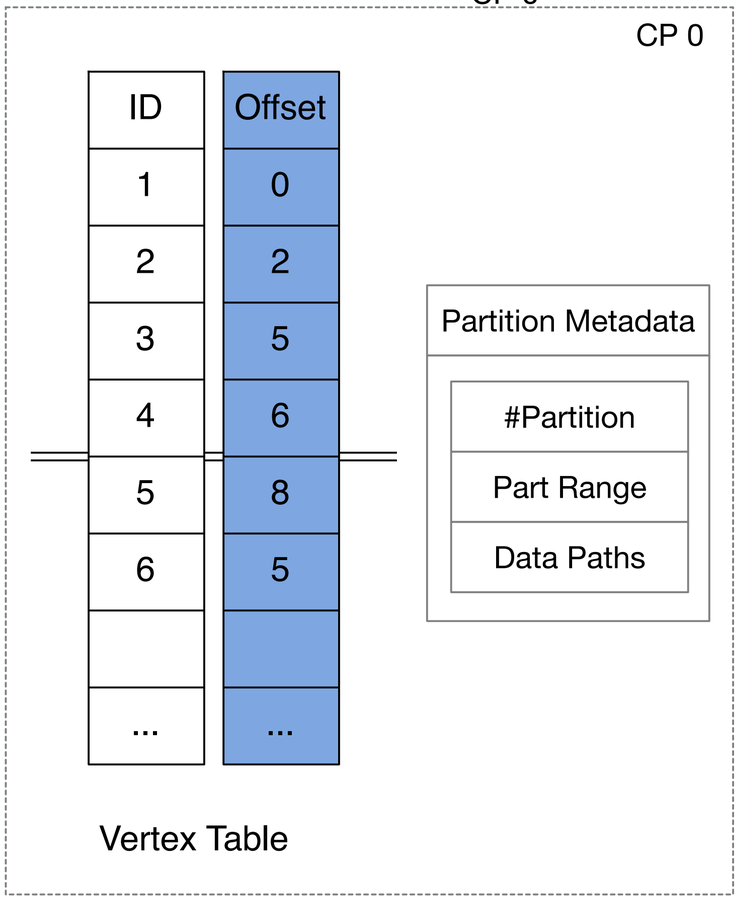

1
10
12
4
17
18
2
13
14
15
3
10
5
6
...
MEM
One Edge Page
One Msg Page
Machine 1
thread 1
thread 2
Machine 2
thread 1
thread 2
ID
Value
Active
Offset
1 1.0 true 0
2 1.0 true 2
3 1.0 true 5
4 1.0 false 6
5 1.0 true 8
6 1.0 true 5
Vertex Table
10
12
13
14
15
10
17
18
Edge Table
MEM
NVM
vertex
msg
1 5
1 2
2 3
3 1
4 2
4 1
5 2
6 12
Msg Table
Build by Computing Algorithms

# Lightweight Distributed Checkpointing

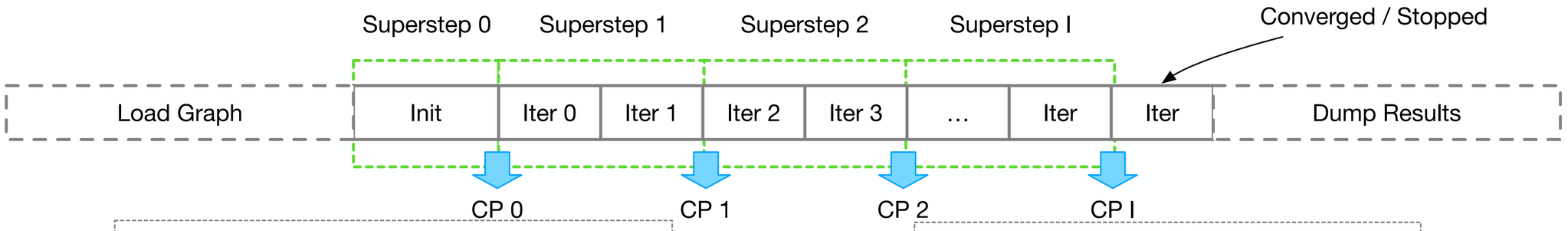


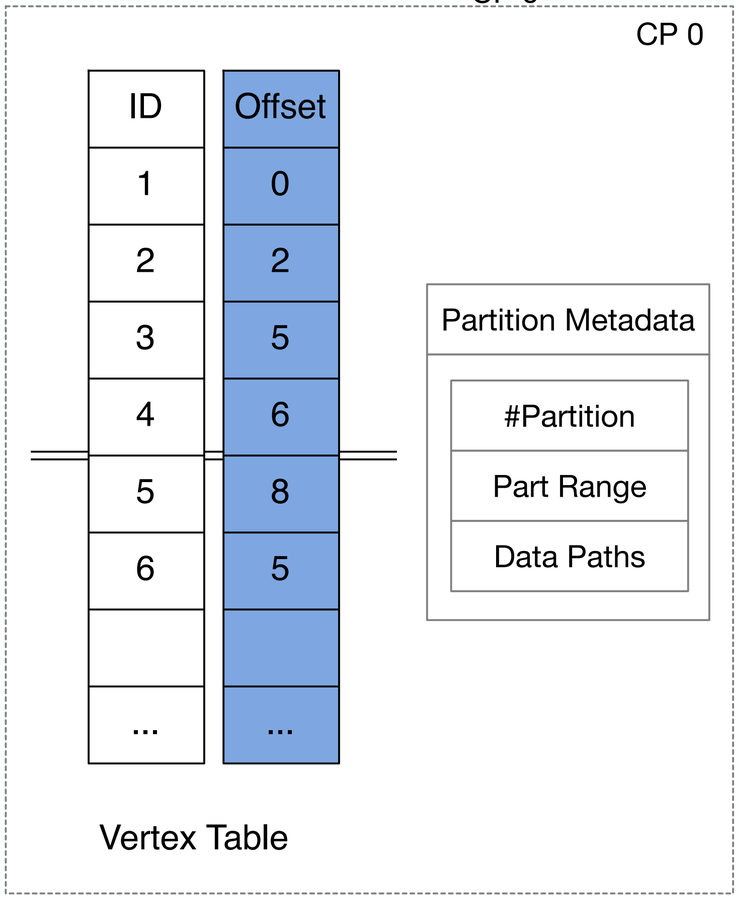
# Lightweight Distributed Checkpointing

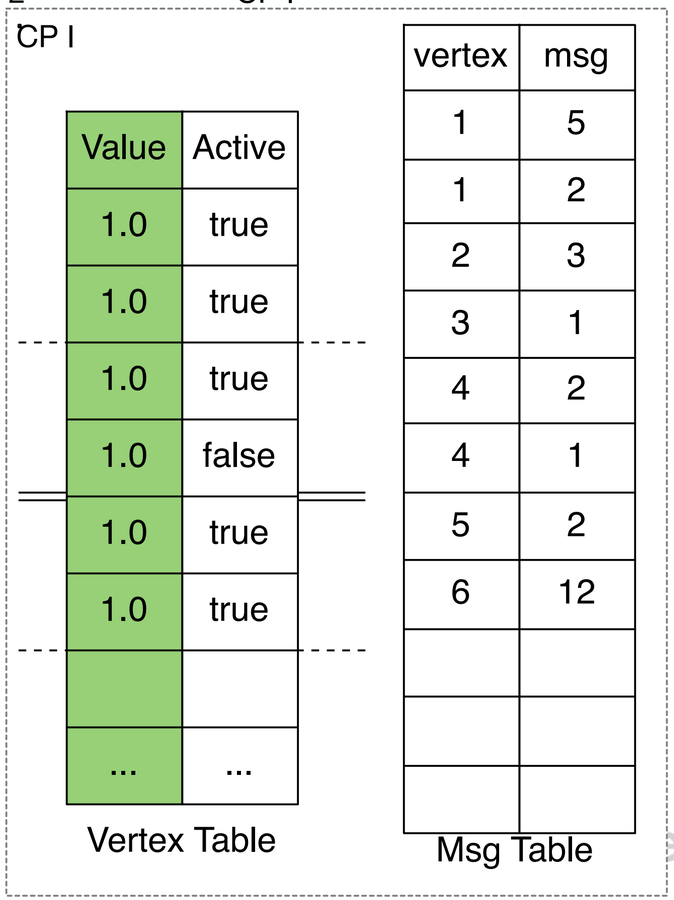


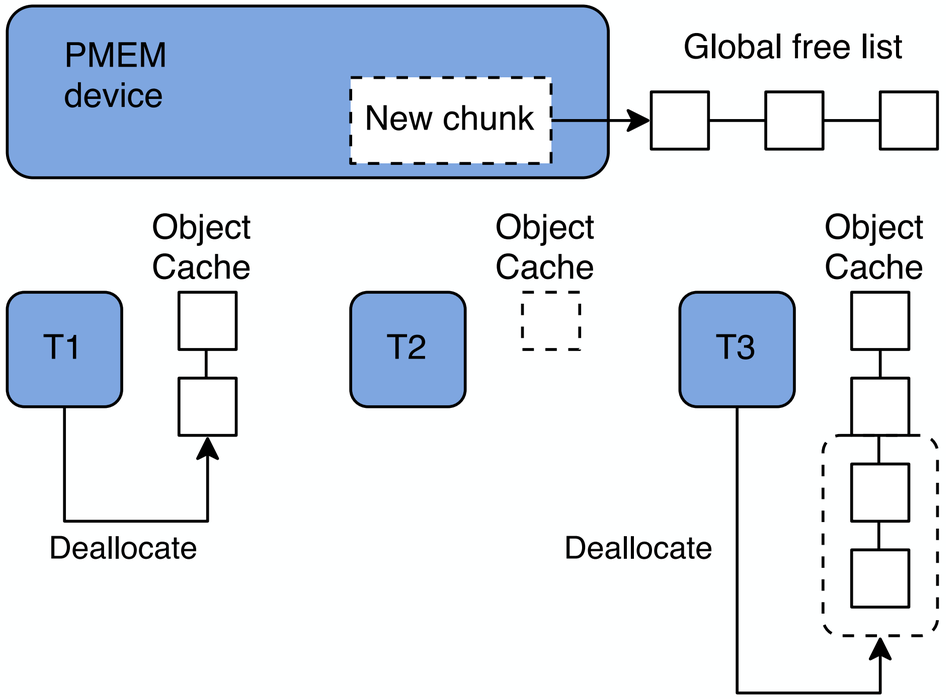
# Lightweight Distributed Checkpointing

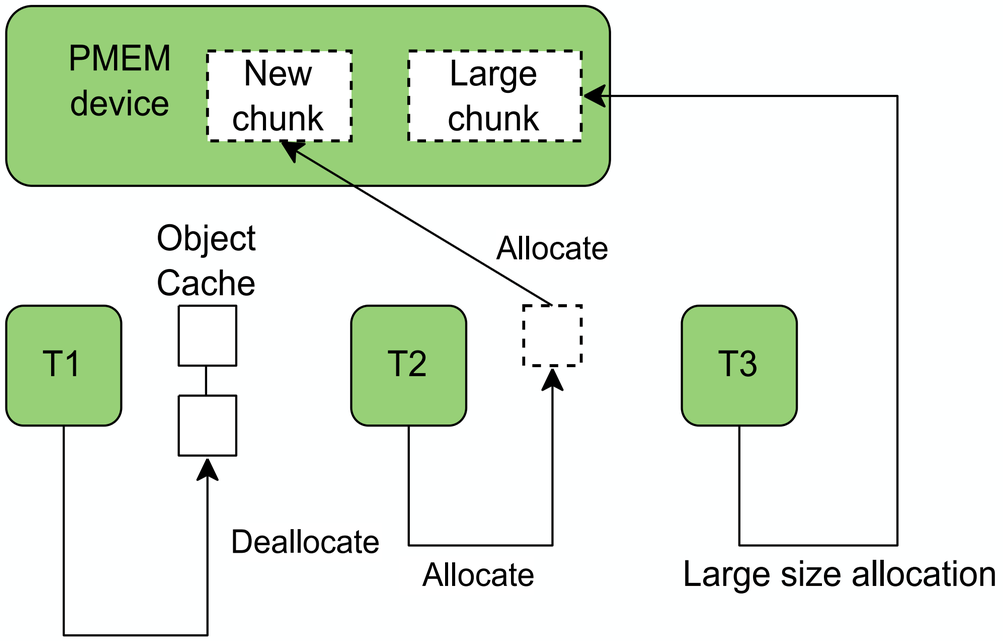


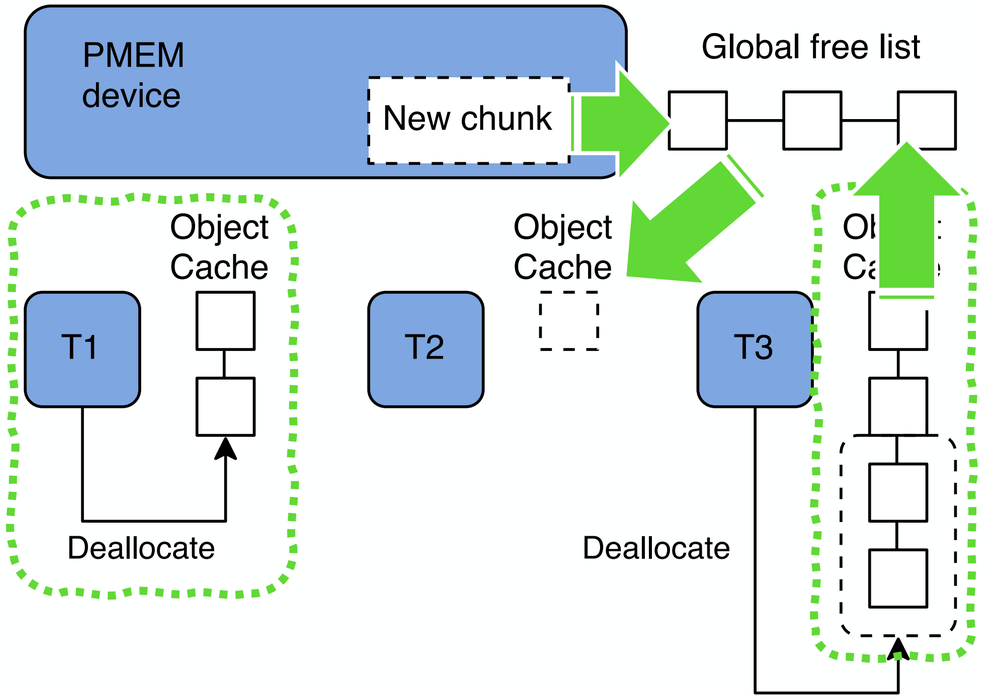



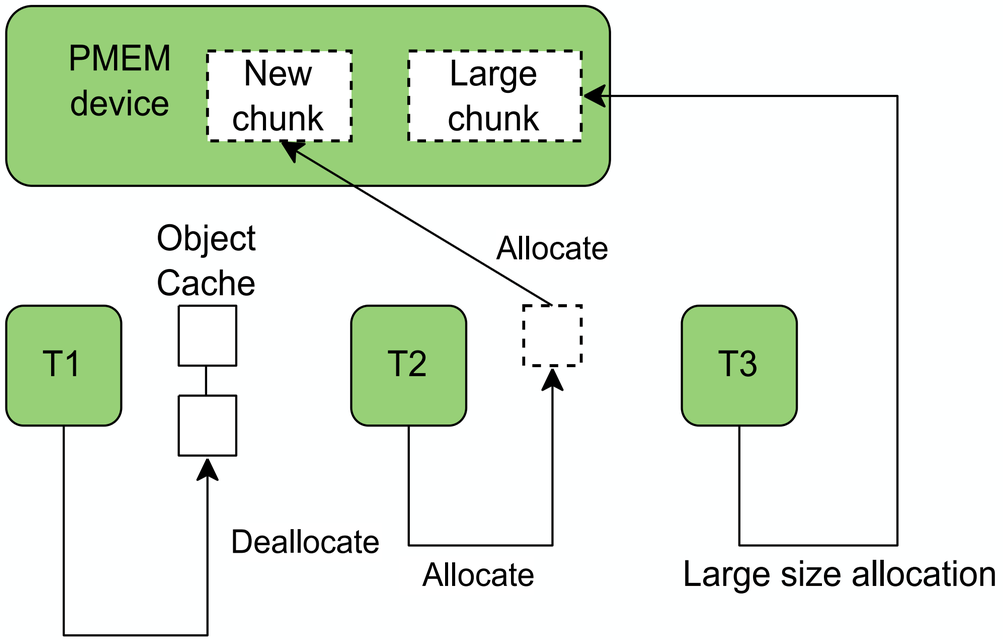
# Dual-Mode Persistent Memory Management



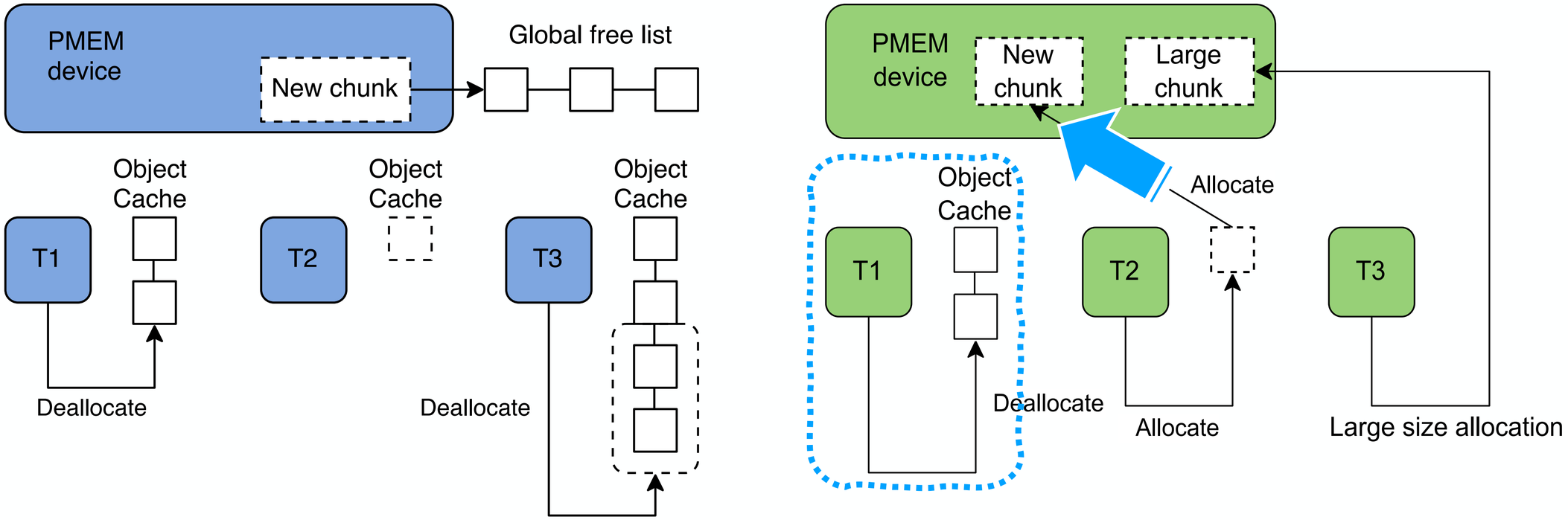



# Dual-Mode Persistent Memory Management



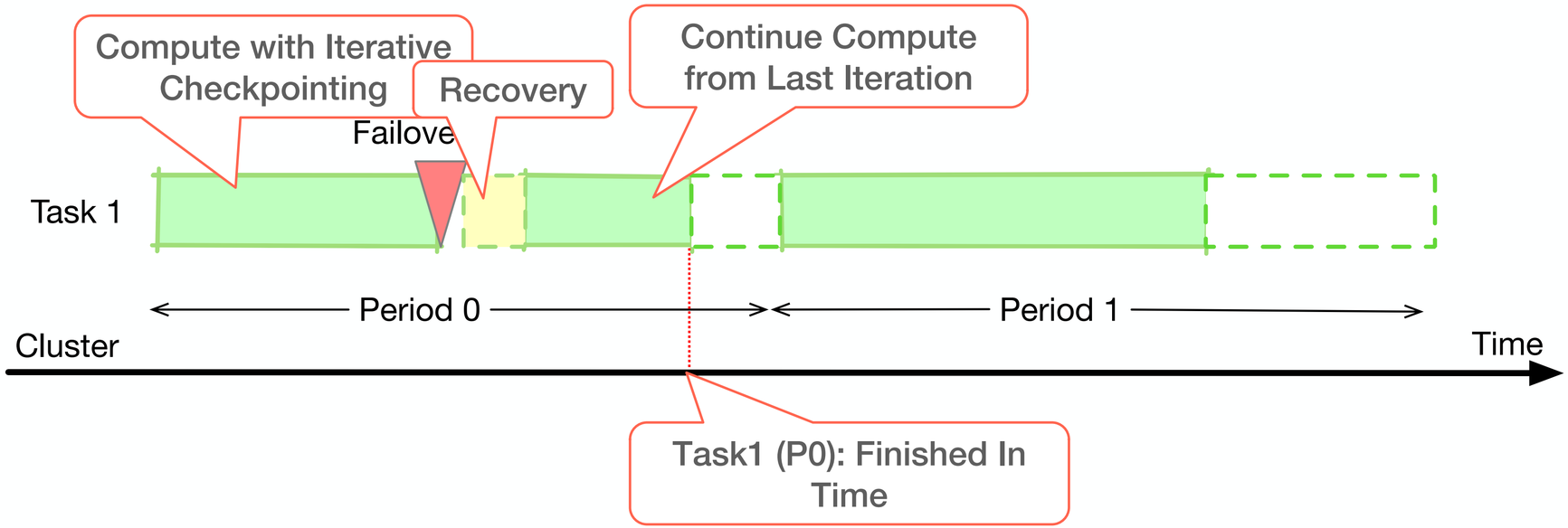



# Dual-Mode Persistent Memory Management



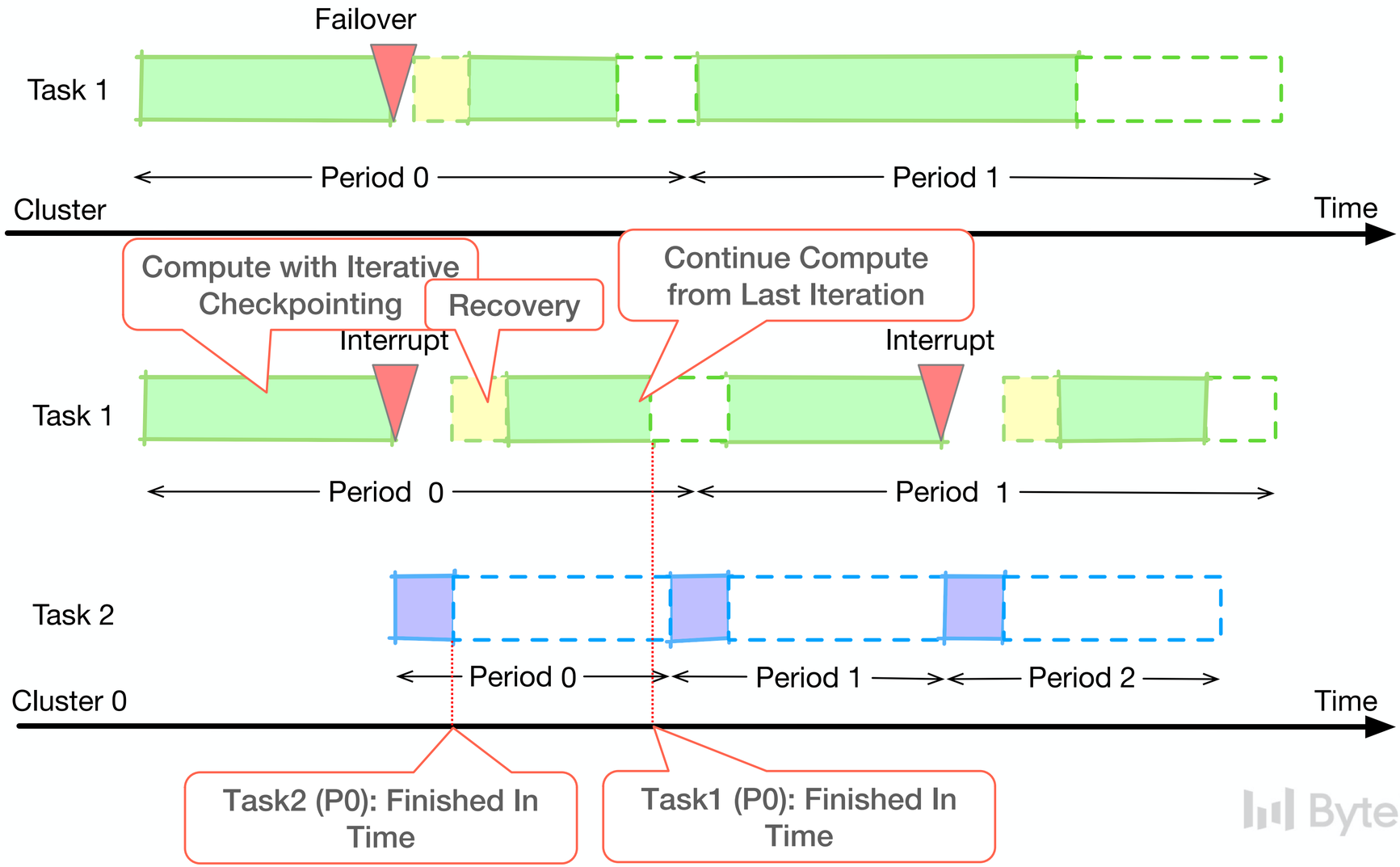
# Our Solutions

ByteGAP : Non-continuous distributed graph computing system based on PMEM



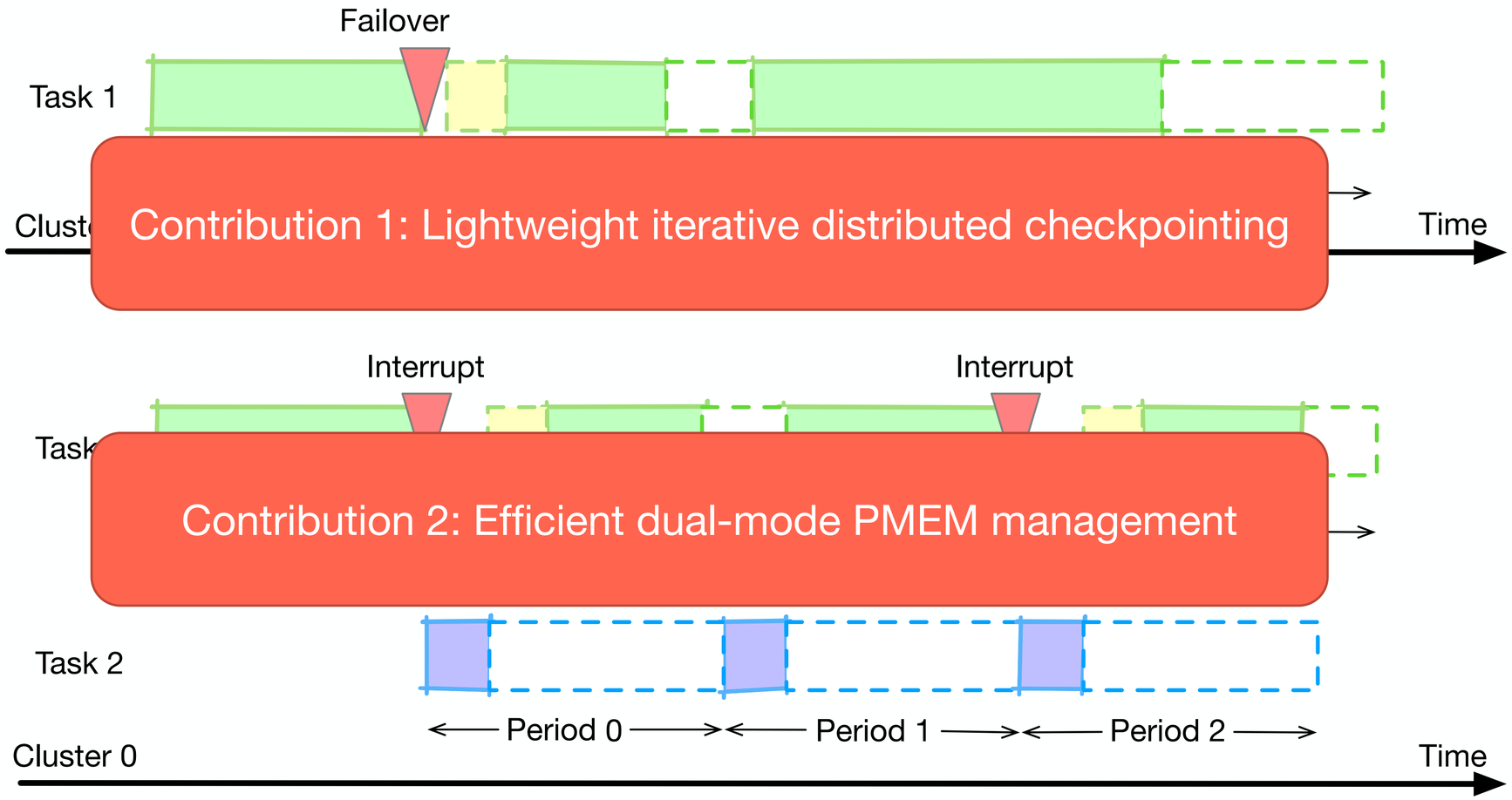
# Our Solutions

ByteGAP : Non-continuous distributed graph computing system based on PMEM



# Our Contributions

ByteGAP : Non-continuous distributed graph computing system based on PMEM



# Evaluations

| Dataset | \|V\| | \|E\| |
|---|---|---|
| Twitter | 41,652,230 | 1,468,364,884 |
| Friendster | 65,608,366 | 1,806,067,135 |
| UK-2007 | 105,896,555 | 3,738,733,648 |
| UK-union | 133,633,040 | 5,475,109,924 |

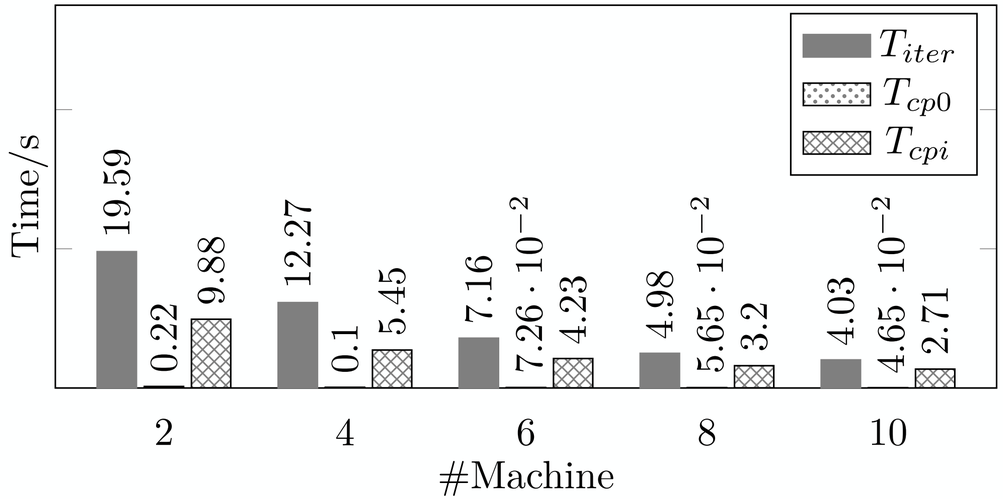
Testbed:
- 10 Machines
- Two Intel Xeon Platinum 8260 CPUs (48 cores)
- 128GB of DRAM
- 320GB NVMe SSD INTEL SSDPE2KX020T8
- 512GB Optane DC PMEM

Baselines:
- Spark 3.0 GraphX[1]

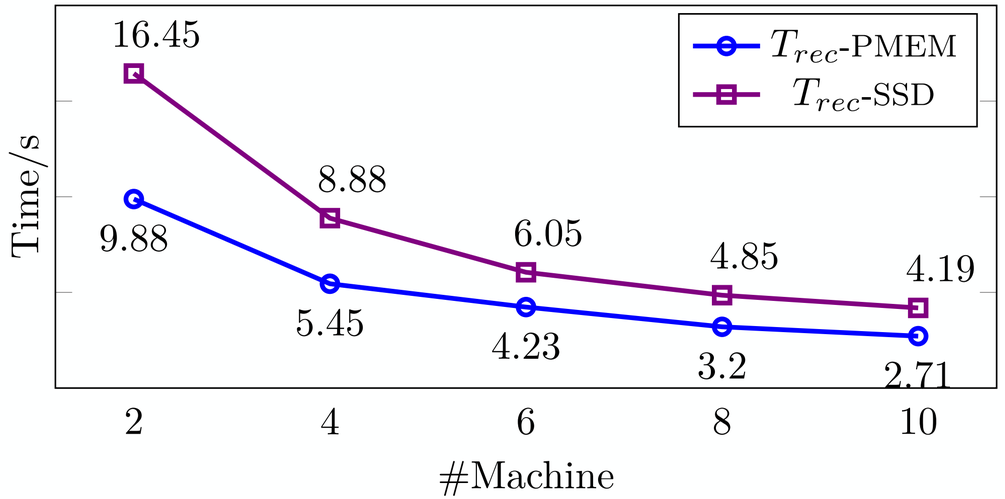
Algorithms:
- PageRank (PR)
- Connected Components (CC)

[1] https://spark.apache.org/graphx/

# Evaluations: Checkpointing

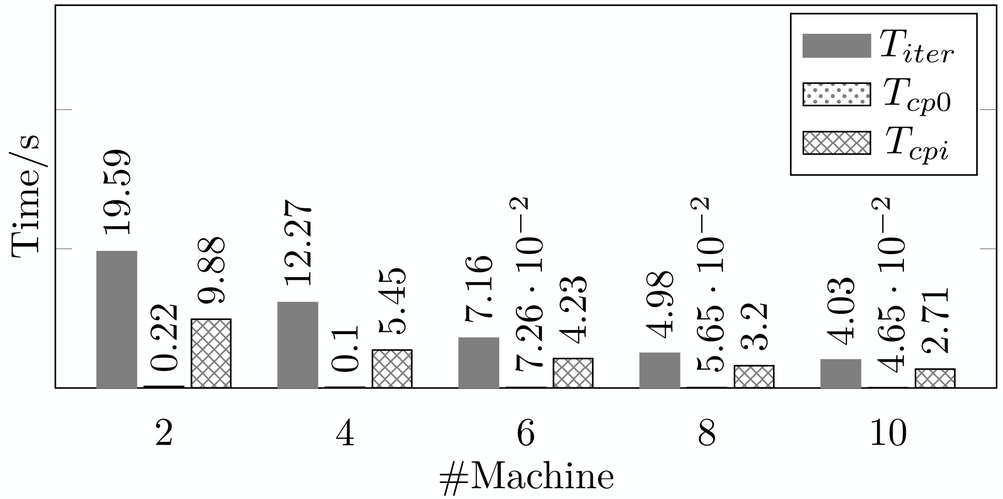



Run-time Breakdown

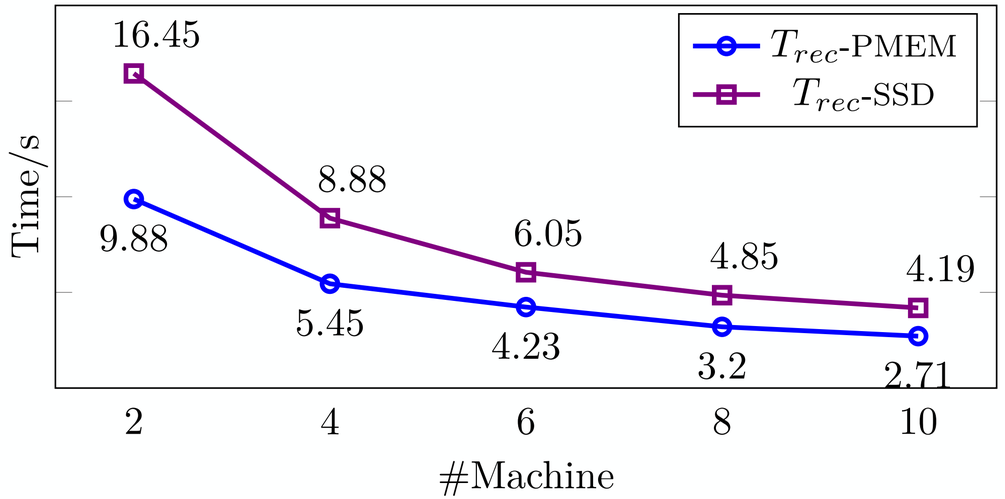



Recovery Time

# Evaluations: Checkpointing

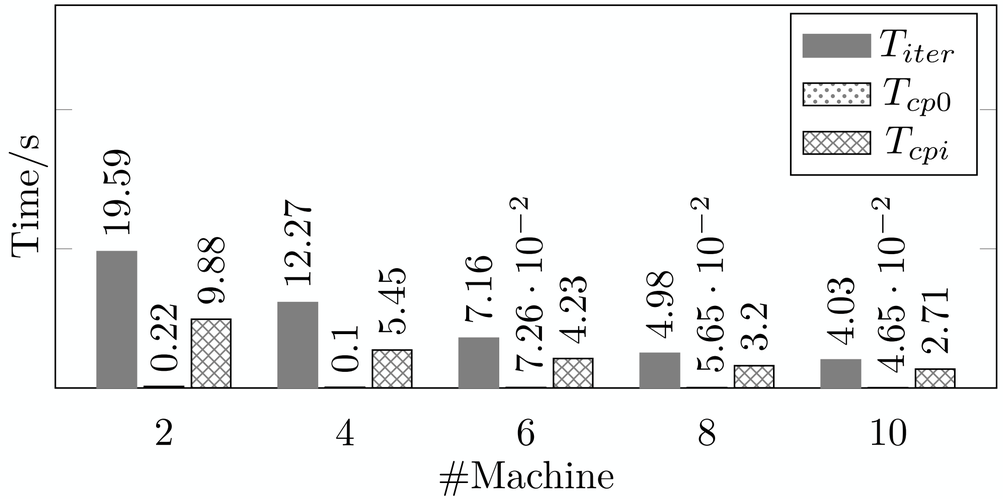


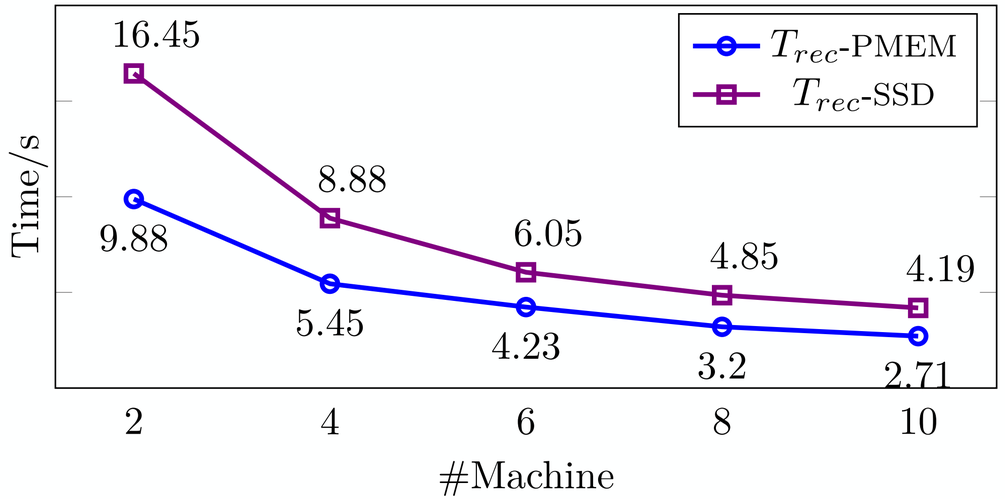

Run-time Breakdown




Recovery Time

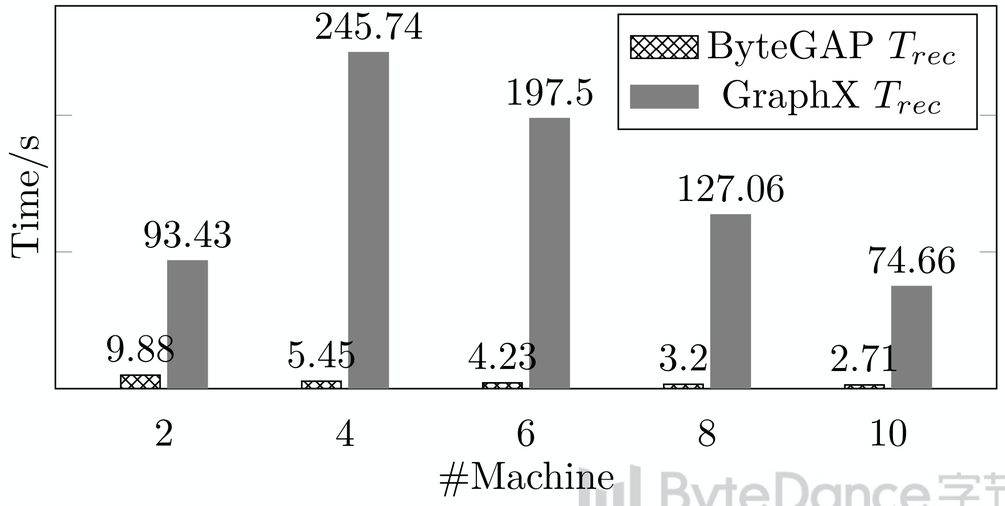
# Evaluations: Checkpointing



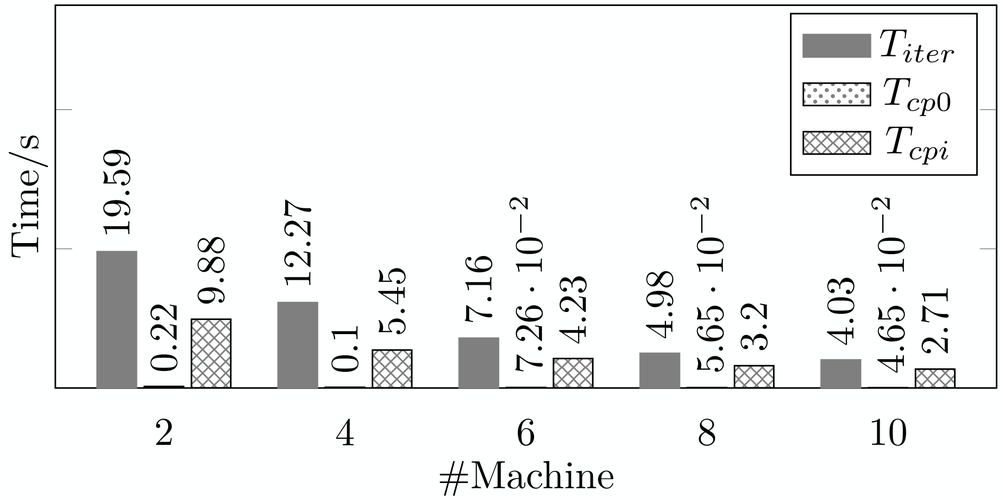



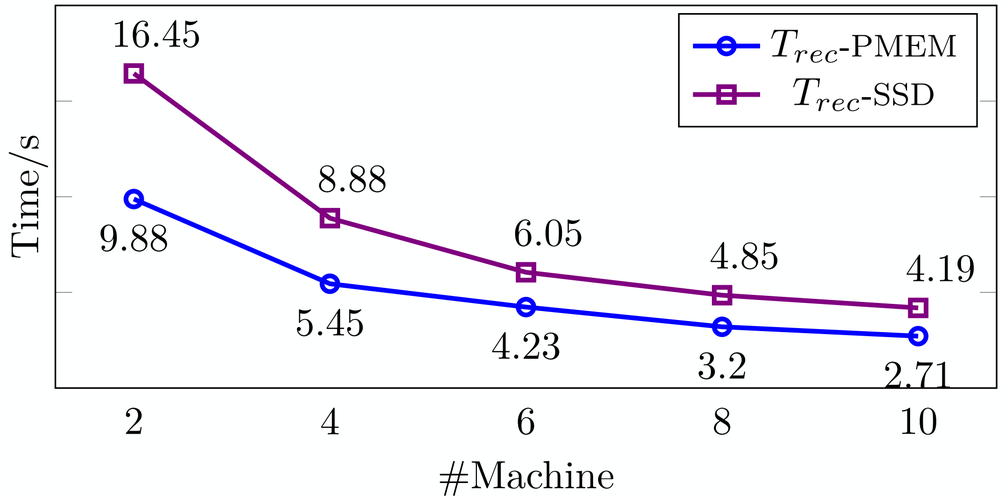

*Compare with GraphX*

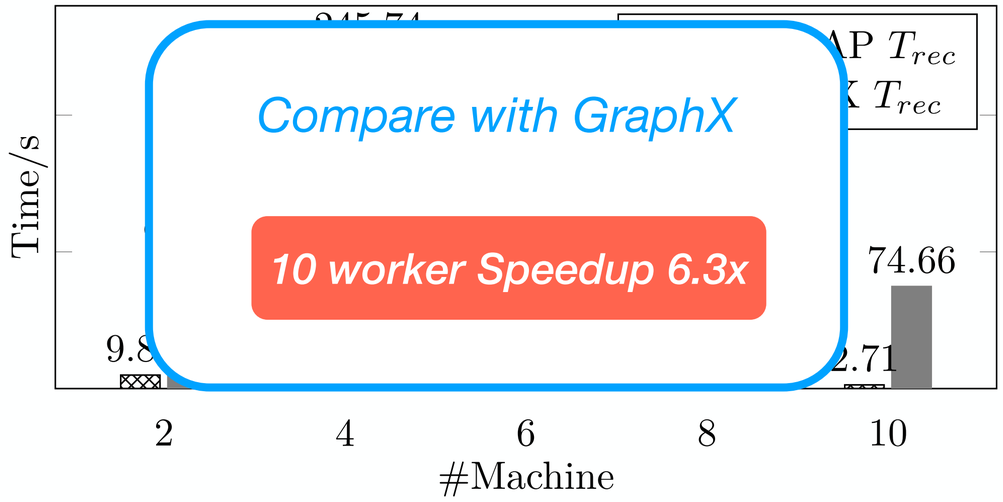
*10 worker Speedup 6.3x*



# Evaluations: Checkpointing

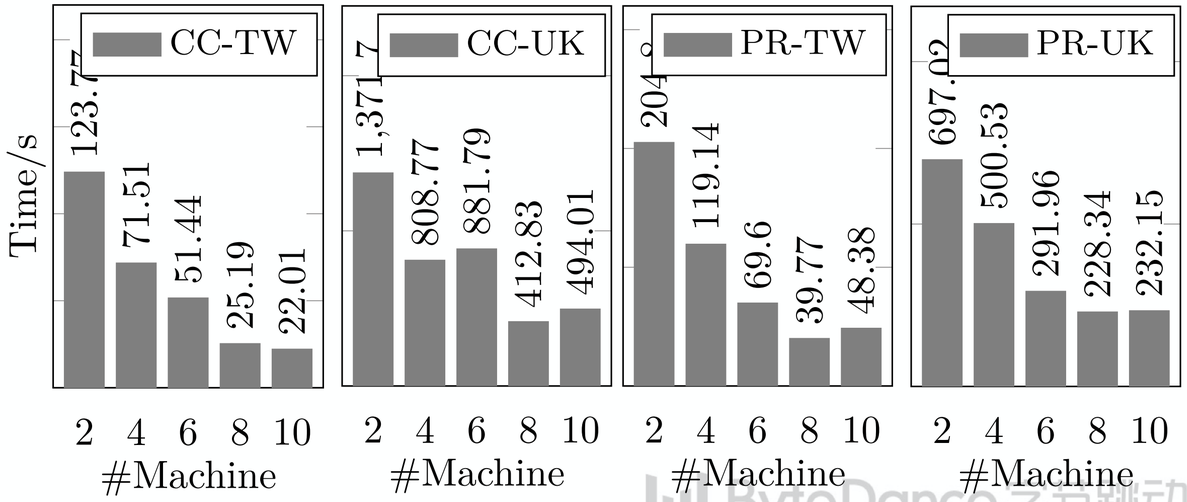



*Compare with GraphX*

*10 worker Speedup 6.3x*




Datasets: Twitter(TW), UKunion(UK)

# Thanks
# Q & A

ByteDance 字节跳动