Dynamic Parameter Allocation in Parameter Servers

Alexander Renz-Wieland[1], Rainer Gemulla[2], Steffen Zeuch[1,3], Volker Markl[1,3]

[1]TU Berlin, [2]University of Mannheim, [3]DFKI

VLDB 2020

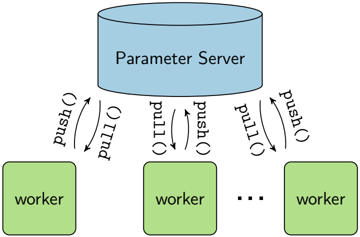
Takeaways

- Key challenge in distributed Machine Learning (ML): communication overhead
- Parameter Servers (PSs)
  - Intuitive
  - Limited support for common techniques to reduce overhead
- How to improve support?
  - Dynamic parameter allocation
- Is this support beneficial?
  - Up to two orders of magnitude faster

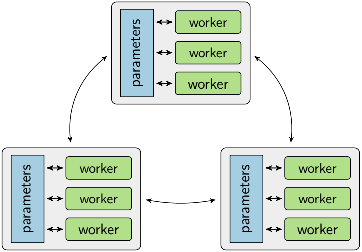
Background: Distributed Machine Learning

- Distributed training is a necessity for large-scale ML tasks
- Parameter management is a key concern
- *Parameter servers* (PS) are widely used

Logical



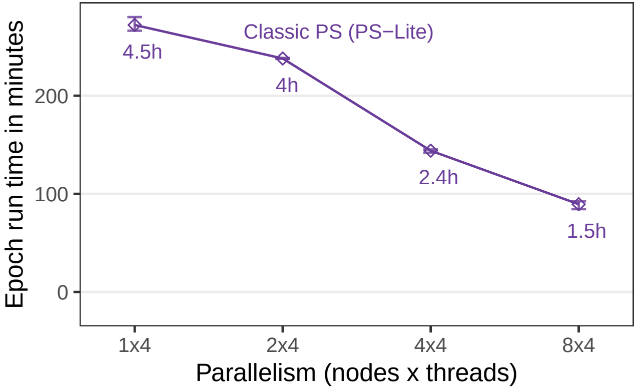
Physical



Problem: Communication Overhead

- Communication overhead can limit scalability
- Performance can fall behind a single node

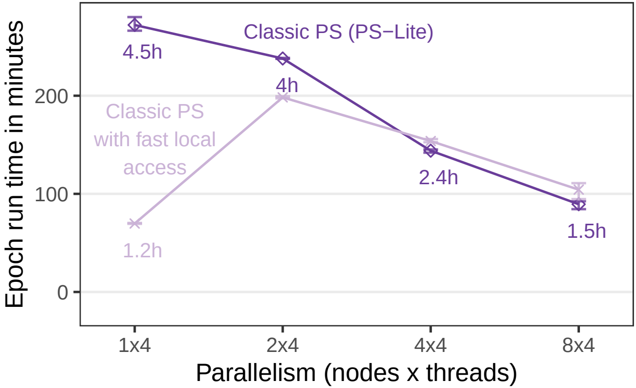
Training knowledge graph embeddings (RESCAL, dimension 100):



Problem: Communication Overhead

- Communication overhead can limit scalability
- Performance can fall behind a single node

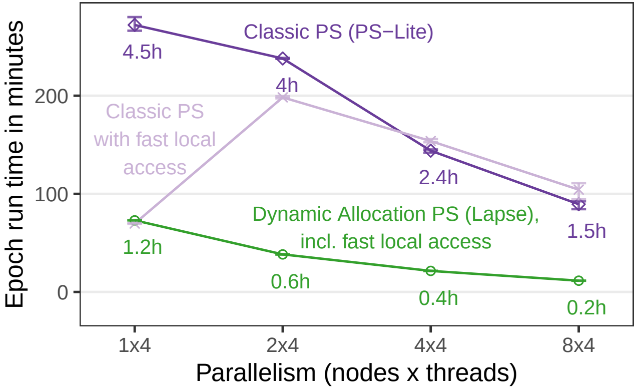
Training knowledge graph embeddings (RESCAL, dimension 100):



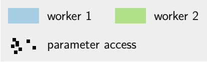
Problem: Communication Overhead

- Communication overhead can limit scalability
- Performance can fall behind a single node

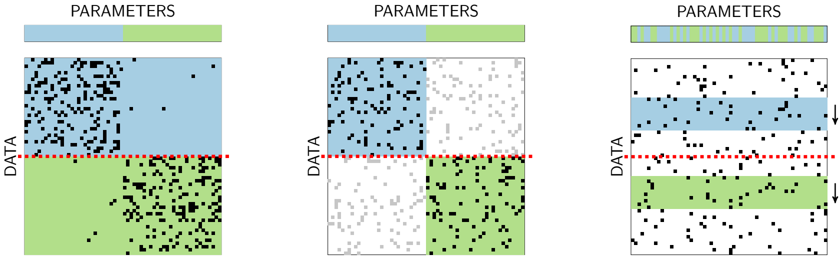
Training knowledge graph embeddings (RESCAL, dimension 100):



How to reduce communication overhead?



- Common techniques to reduce overhead:

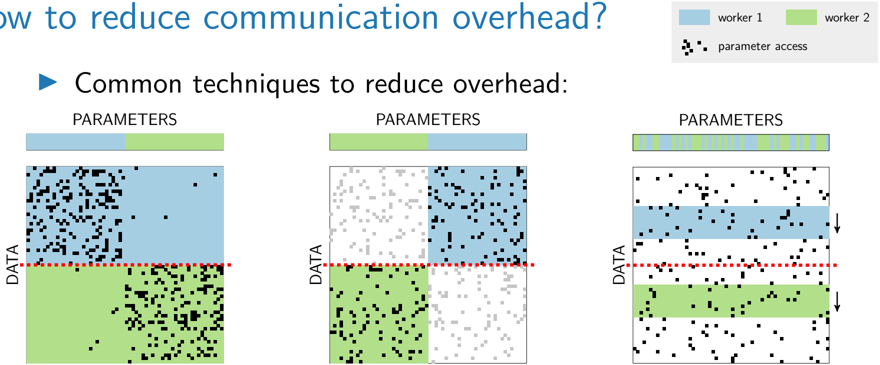


Data clustering　　Parameter blocking　　Latency hiding

- Key is to avoid remote accesses
- Do PSs support these techniques?
  - Techniques require local access at different nodes over time
  - But PSs allocate parameters statically

How to reduce communication overhead?

- Common techniques to reduce overhead:



Data clustering | Parameter blocking | Latency hiding

- Key is to avoid remote accesses
- Do PSs support these techniques?
  - Techniques require local access at different nodes over time
  - But PSs allocate parameters statically

Dynamic Parameter Allocation

- What if the PS could allocate parameters dynamically?

```
Localize(parameters)
```

- Would provide support for
  - Data clustering ✓
  - Parameter blocking ✓
  - Latency hiding ✓
- We call this **dynamic parameter allocation**

The Lapse Parameter Server

- Features
  - Dynamic allocation
  - Location transparency
  - Retains sequential consistency

PS per-key consistency guarantees (for synchronous operations)

	Classic	Lapse	Stale
Eventual	✓	✓	✓
PRAM	✓	✓	✓
Causal	✓	✓	×
Sequential	✓	✓	×
Serializability	×	×	×

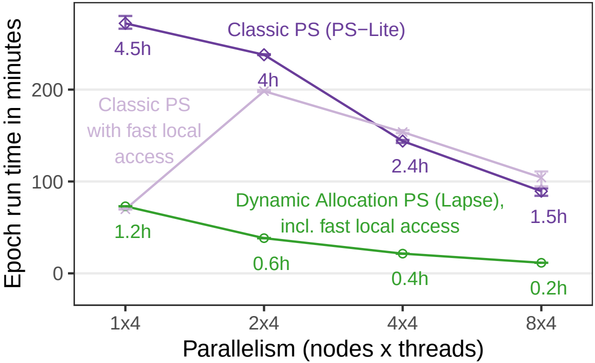
- Many system challenges (see paper)
  - Manage parameter locations
  - Route parameter accesses to current location
  - Relocate parameters
  - Handle reads and writes during relocations
  - All while maintaining sequential consistency

Experimental study

Tasks: matrix factorization, knowledge graph embeddings, word vectors

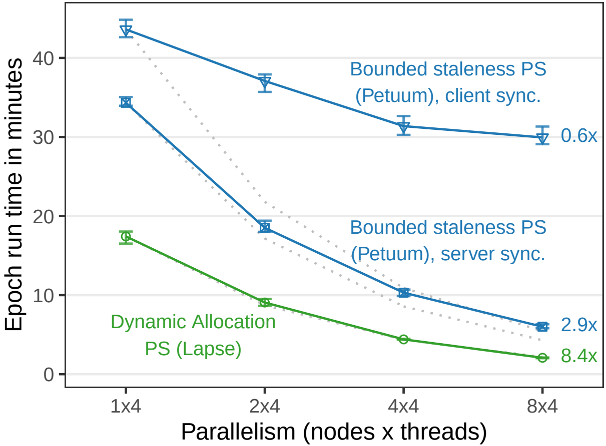
Cluster: 1–8 nodes, each with 4 worker threads, 10 GBit Ethernet

1. Performance of Classic PSs
   - 2–8 nodes barely outperformed 1 node in all tested tasks
2. Effect of dynamic parameter allocation
   - 4–203x faster than a Classic PSs, up to linear speed-ups



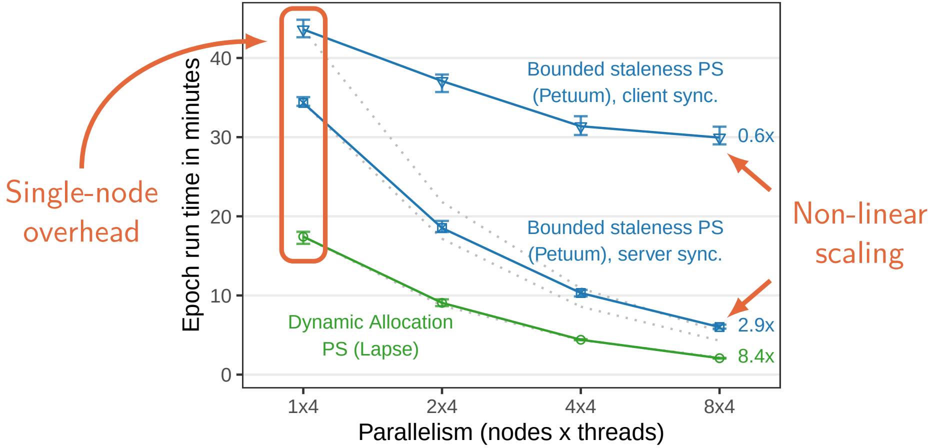
Comparison to Bounded Staleness PS

- Matrix factorization (matrix with 1b entries, rank 100)
- Parameter blocking



Comparison to Bounded Staleness PS

- Matrix factorization (matrix with 1b entries, rank 100)
- Parameter blocking



Experimental study

Tasks: matrix factorization, knowledge graph embeddings, word vectors
Cluster: 1–8 nodes, each with 4 worker threads, 10 GBit Ethernet

1. Performance of Classic PSs
   - 2–8 nodes barely outperformed 1 node in all tested tasks
2. Effect of dynamic parameter allocation
   - 4–203x faster than a Classic PSs, up to linear speed-ups
3. Comparison to bounded staleness PSs
   - 2–28x faster and more scalable
4. Comparison to manual management
   - Competitive to a specialized low-level implementation
5. Ablation study
   - Combining fast local access and dynamic allocation is key

Dynamic Parameter Allocation in Parameter Servers

- Key challenge in distributed Machine Learning (ML): communication overhead
- Parameter Servers (PSs)
  - Intuitive
  - Limited support for common techniques to reduce overhead
- How to improve support?
  - Dynamic parameter allocation
- Is this support beneficial?
  - Up to two orders of magnitude faster
- Lapse is open source: `https://github.com/alexrenz/lapse-ps`