~~SETTING FOUNDATIONS~~

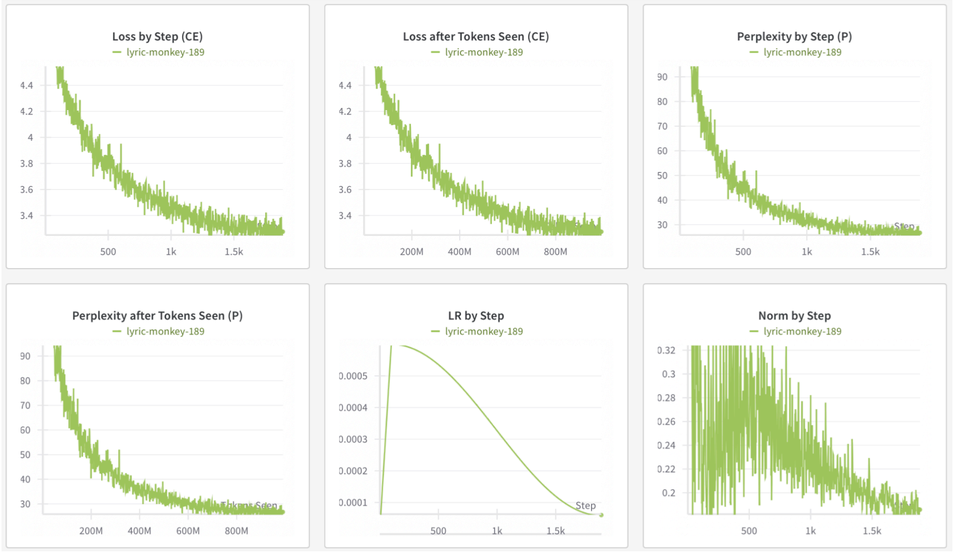
Going Distributed

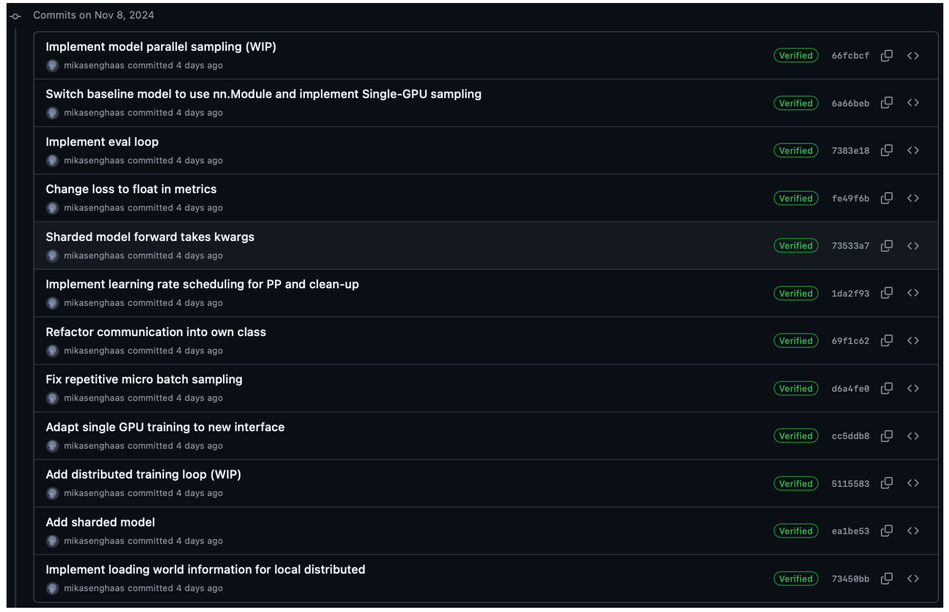
MIKA
SENGHAAS

EPFL

Next Steps

1. Verify 1-GPU Baseline
   - NanoGPT setup
   - (GPT-2 (140m) on FineWebEdu on A100)
2. Implement traditional PP → Minimal (200ish) PP Gist
3. Implement SWARM → Yandex SWARM + PI ZeroBand





- GPT-2 1BT (Test Loss: ~3.1)
- AFAB Pipeline Parallel Implementation

Train GPT2 124M on 1B tokens from FinewebEdu (Single GPU)

- In-/post-training evaluation
- Sampling
- Checkpointing (Local/ W&B)
- LR Scheduling
- Mixed Precision Training

Test CE Loss ~3.1

Implement AFAB Pipeline Parallel

- Sharded model with custom forward/ backward pass
- *torch.distributed* for comm of activations and gradients
- In-/post-training evaluation
- *Sampling*
- *Checkpointing (Local/ W&B)*
- *Mixed Precision Training*

Minimal implementation - loads of reusable components

```
(base) root@6c57594e146a:/workspace/swarm# find . -name "*.py" | xargs wc -l
    0 ./src/__init__.py
  260 ./src/train/baseline.py
  300 ./src/train/pipeline.py
   48 ./src/comm.py
   96 ./src/config.py
  117 ./src/logger.py
  192 ./src/metrics.py
   76 ./src/model.py
  189 ./src/utils.py
   29 ./src/world.py
 1307 total
```

Next Steps

1. ~~Verify 1 GPU baseline~~
2. Implement traditional PP → Minimal (200ish) PP Gist
3. Implement SWARM → Yandex SWARM + PI ZeroBand

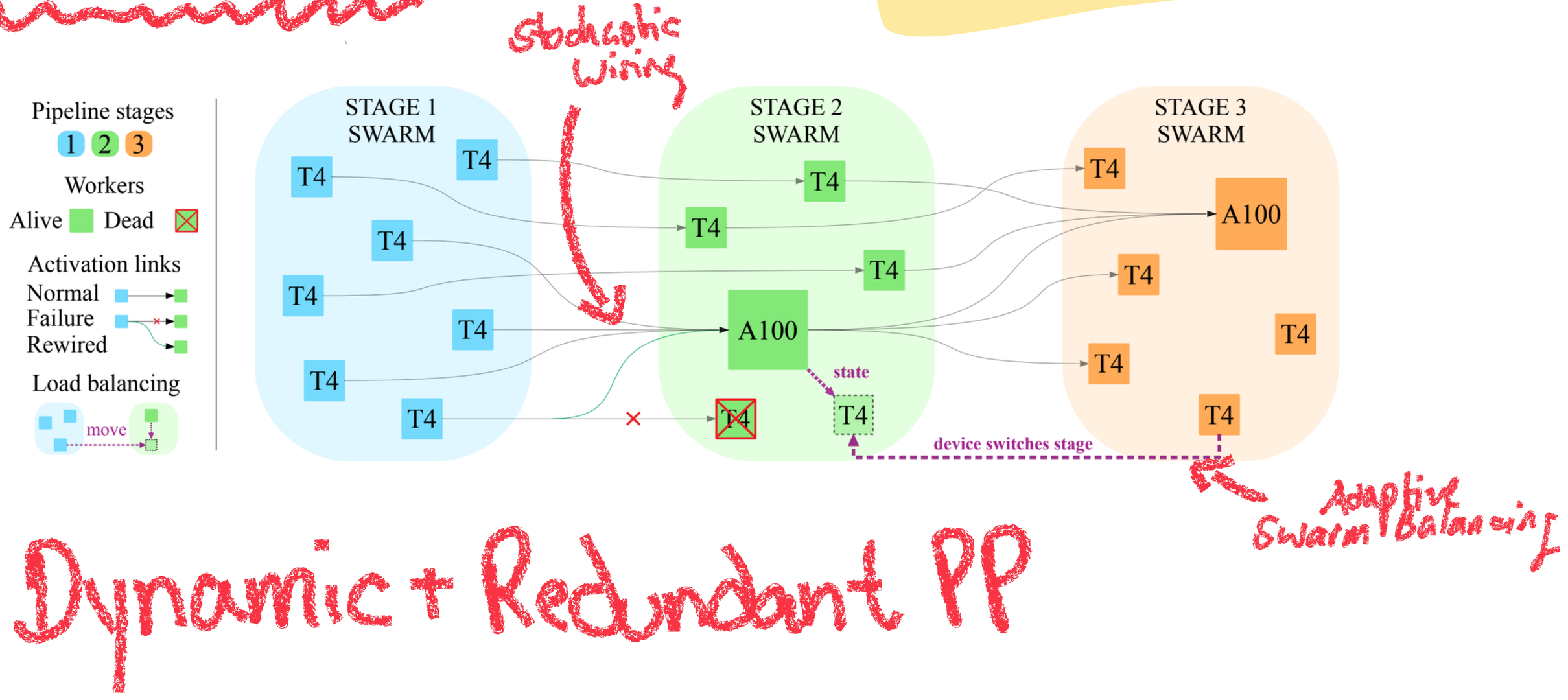
NanoGPT setup → (GPT-2 (140m) on FineWebEdu on A100)

- Allow for truly distributed workers (IPv4+port, instead of local cuda worker -> might have to switch communication backedn)
- Implement missing functionality (remote logging, sampling, MP)
- Benchmark speed and convergence

References

SWARM

Remember?



Dynamic + Redundant PP