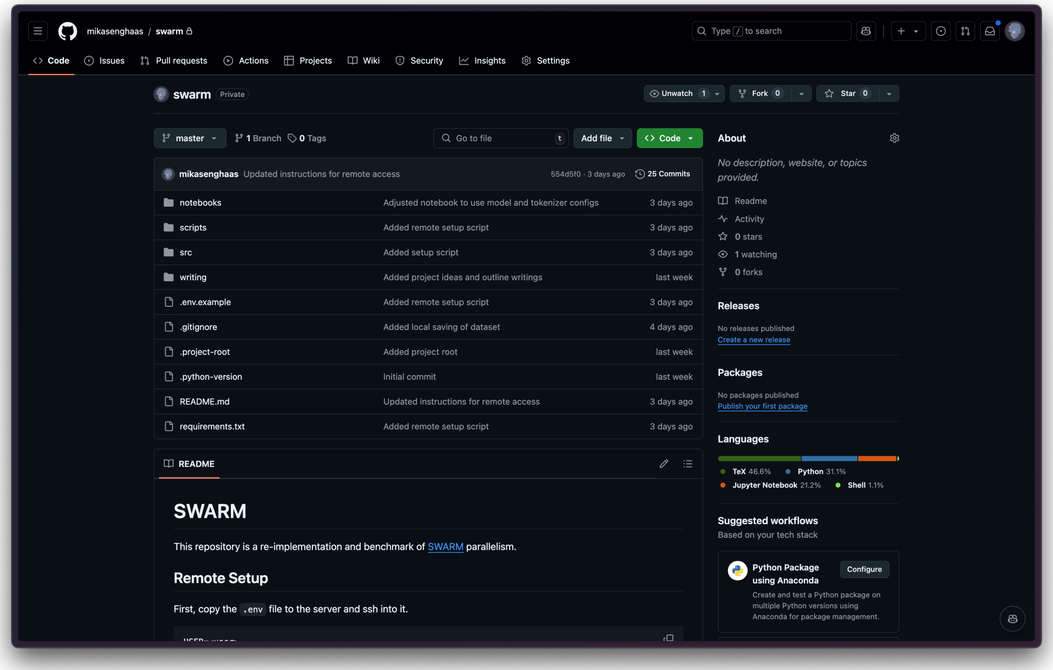
BATTLE-TESTING SWARM

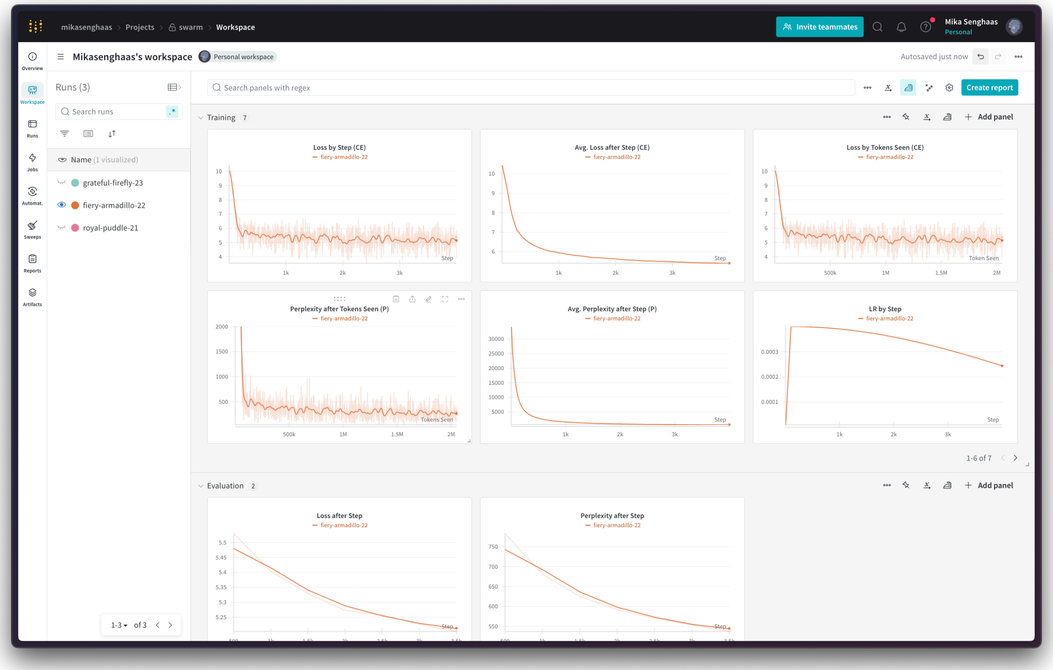
SETTING FOUNDATIONS

MIKA SENGHAAS

EPFL

Implementing Single GPU



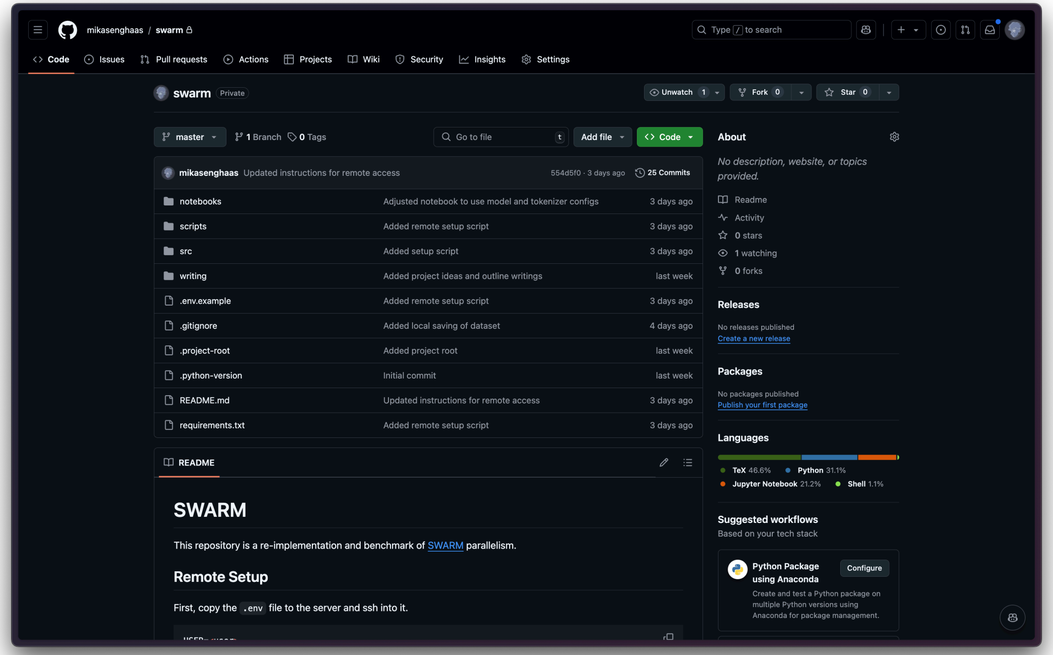


Weights & Biases

Implementing Single GPU

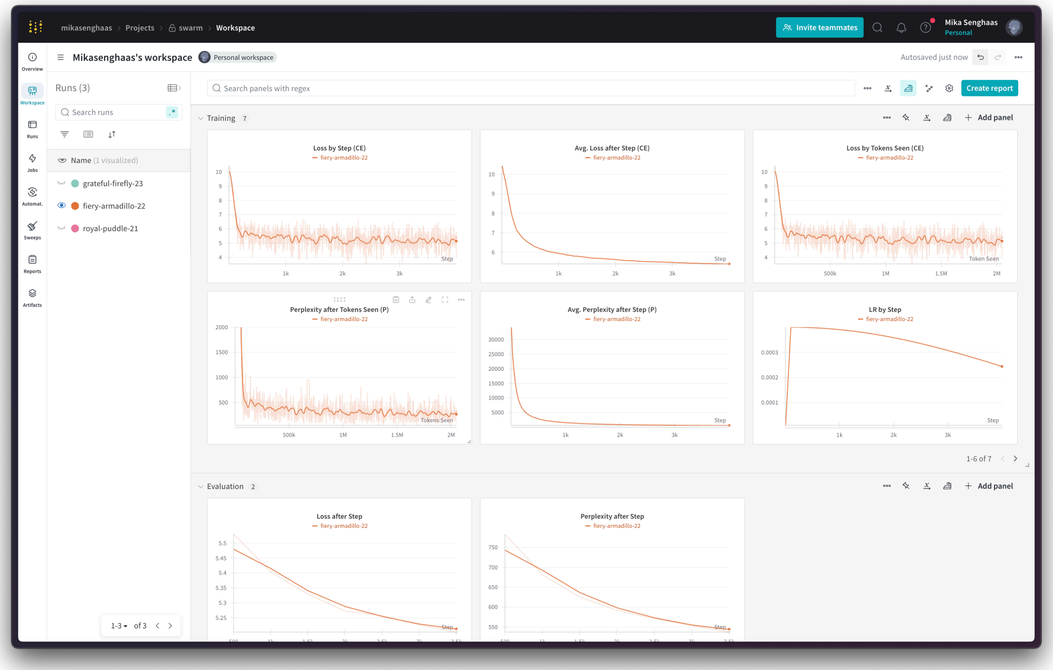
```
[ I ] ~/e/c/p/swarm master ❯ git ls-files | grep '\.py' | xargs wc -l
      1 .python-version
      0 src/__init__.py
     98 src/config.py
    100 src/logger.py
    161 src/metrics.py
    185 src/train/base.py
    175 src/utils.py
    720 total
```

→ 720 LOC / PureTorch



- Minimal
- HF Data/Model (Dummy: Wikitext 2, Llama2 14M)
- Console / W+B Logging
- Advanced config (PyDantic)
- Cosine LR scheduler
- Mixed precision training (on CUDA)
- Gradient Accumulation
- Custom Metrics (+ Logging)
- Easy remote access → Integrates w/ pl Compute

Implementing Single GPU



Weights & Biases

Metrics

- Step
- CE Loss
- Perplexity
- #Tokens (#Examples)
- LR
- Time

Also evaluate

Computed

+Throughput (T/s)

+ W&B Perf. Metrics (GPU Mem./ Utilisation/ Temp./...)

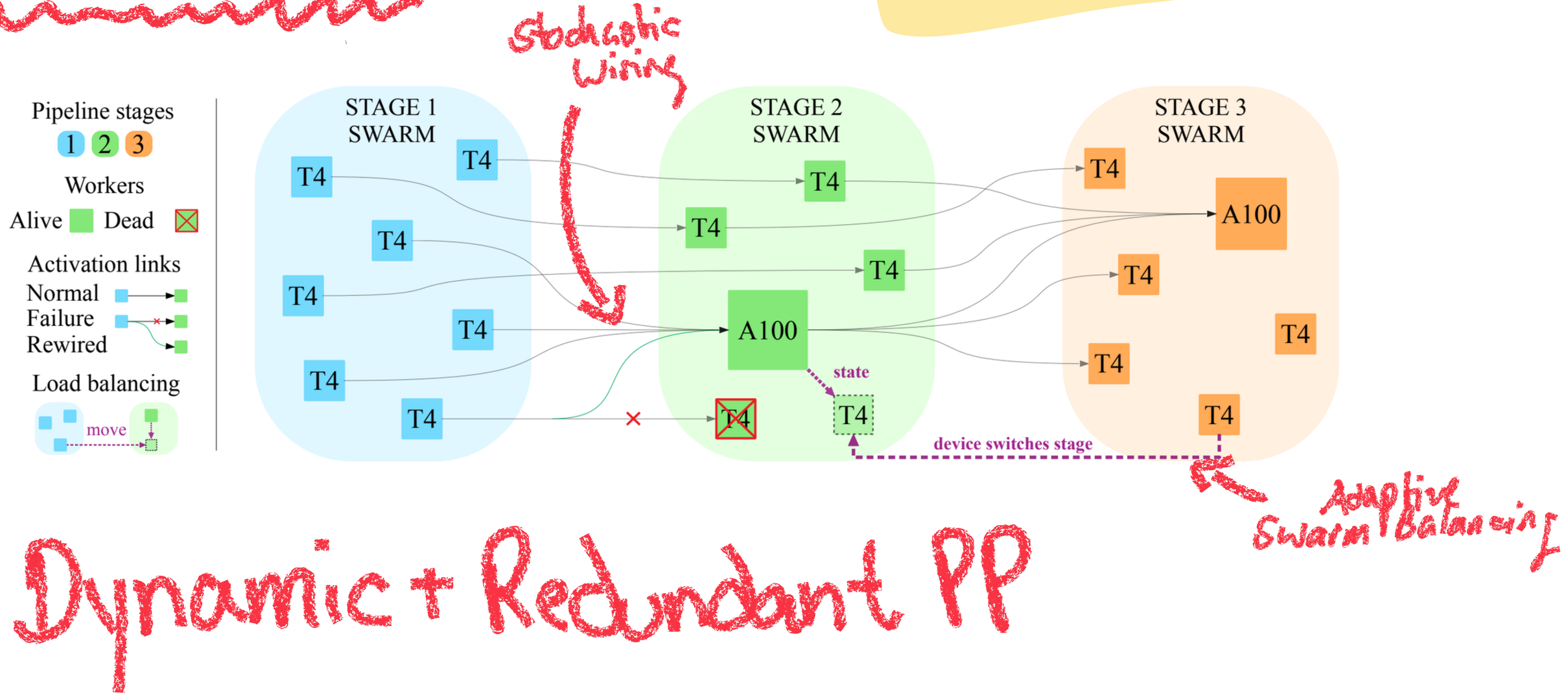
Next Steps

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

References

SWARM

Remember?



Dynamic + Redundant PP

Decentralised Setting

1. Heterogeneous Hardware
2. Heterogeneous Network
3. Reliability
4. Scale

↳ On-/Off Ramping

↳ Infinite Model + Data Scale