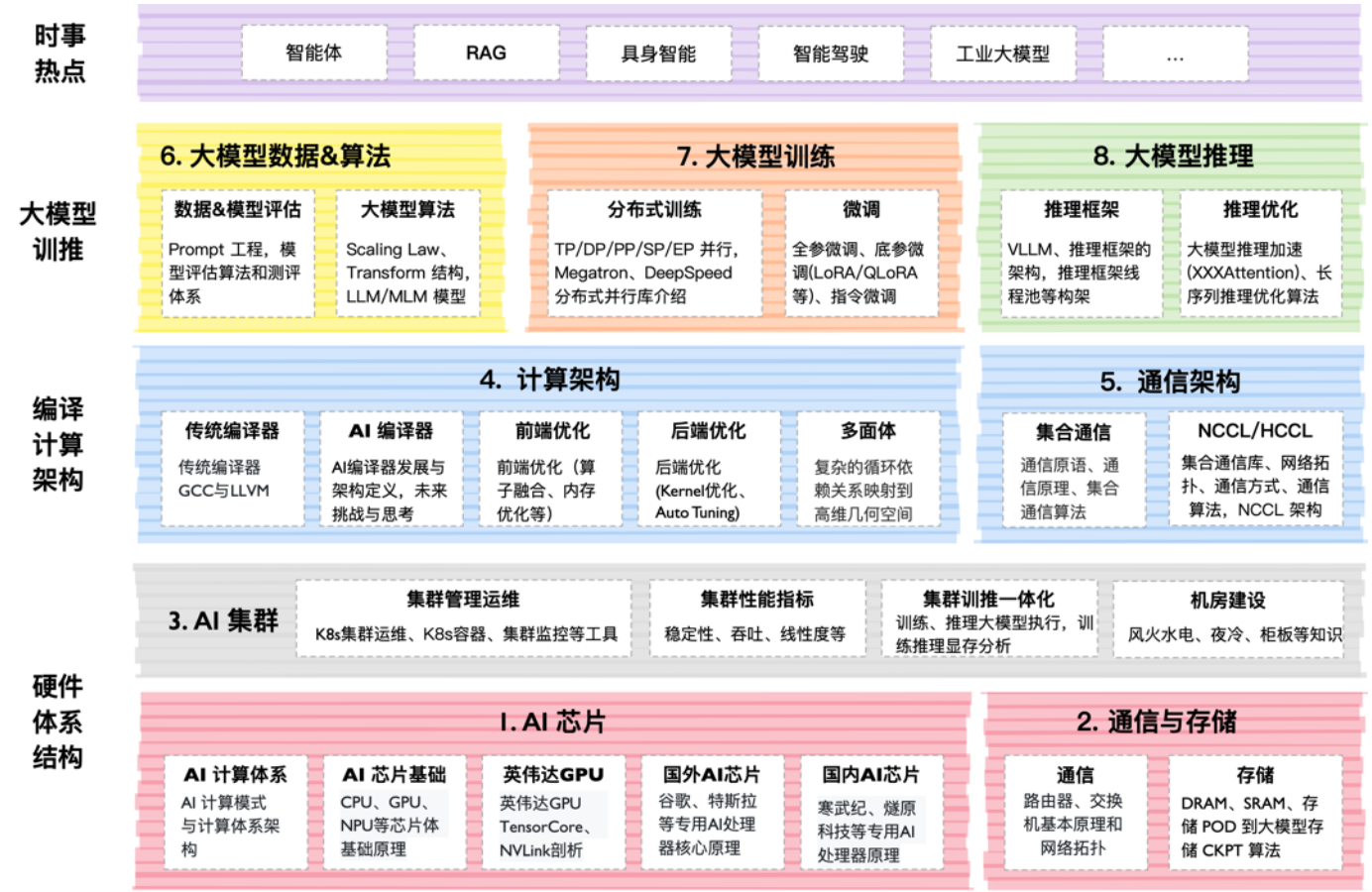
AI 集群
测试方案

ZOMI

Question?

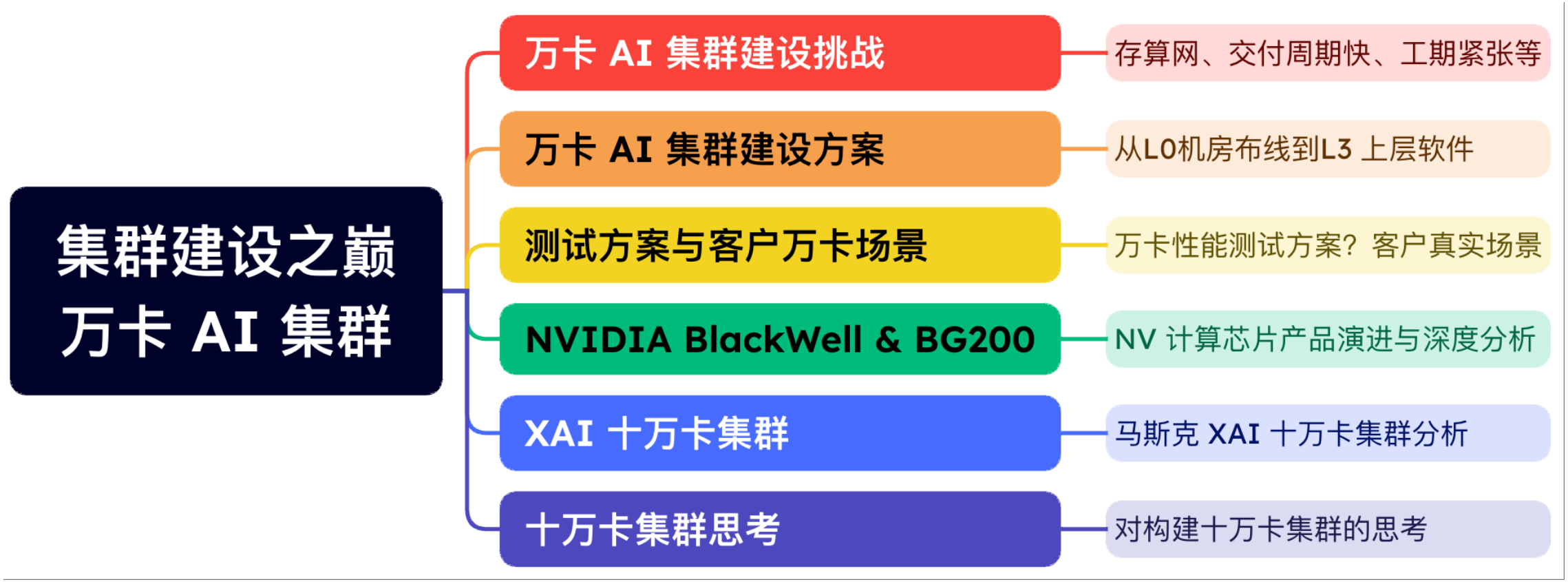
- 交付万卡集群，需要做什么?
- 搭好硬件之后，分配 IP，就给算法同事自己用?

Content github.com/Infrasys-AI/AIInfra



GitHub github.com/Infrasys-AI/AIInfra book infrasys-ai.github.io

Content


集群建设之巅
万卡 AI 集群
万卡 AI 集群建设挑战
存算网、交付周期快、工期紧张等
万卡 AI 集群建设方案
从L0机房布线到L3 上层软件
测试方案与客户万卡场景
万卡性能测试方案？客户真实场景
NVIDIA BlackWell & BG200
NV 计算芯片产品演进与深度分析
XAI 十万卡集群
马斯克 XAI 十万卡集群分析
十万卡集群思考
对构建十万卡集群的思考

Content

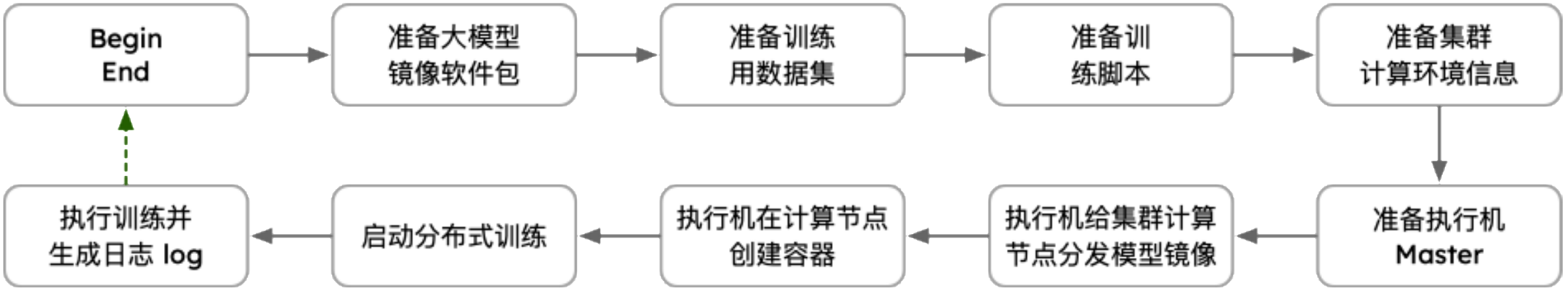
1. 测试步骤（测试准备、测试原则、测试步骤）
2. 性能测试方案（模型、性能、线性度、集合通讯）
3. 总结与思考（问题焦点 + 具体措施 + 量化目标）

01

测试步骤

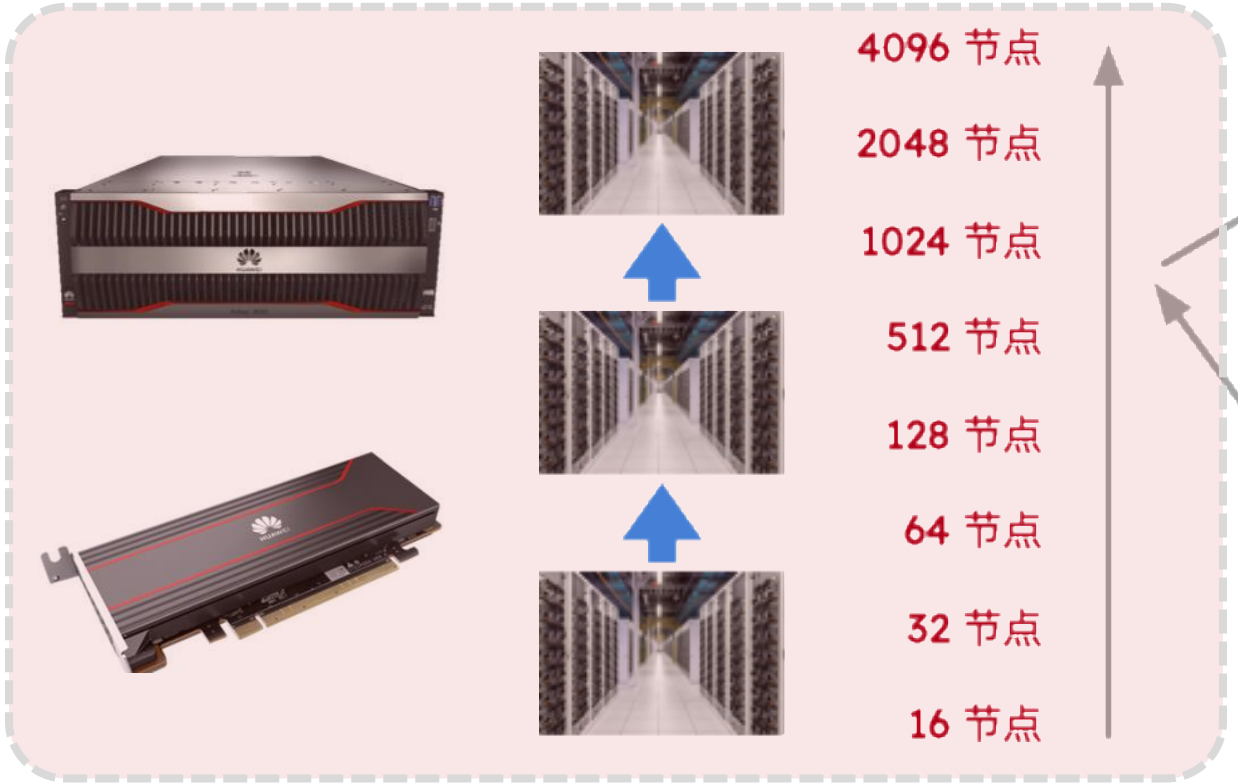
测试前准备

- **Step1：**准备大模型镜像，包含模型文件 + 脚本 + CUDA/CANN + Megatron 等相关软件。
- **Step2：**准备数据集，因为面向集群性能和长稳摸高测试，可直接使用模型开源数据集。
- **Step3：**准备模型训练使用脚本，明确脚本中关于模型训练相关的超参，特别是并行策略。



测试策略原则

- 测试任务由小到大，由易到难，先功能再性能，先测峰值再测长稳。
- 先从小规模集群开始罚试，由小到大，稳步攀升到万卡。



NPU 集群 稳步攀升

GPU 集群 基线拉齐

做好故障冗余备份

- **备份方式：**进行 2K 节点测试时，先准备 2K+X 节点，X 作为备用节点，当 2K 节点出现故障节点时，立即使用 X 备用节点进行替换。既不影响 2K 节点训练测试，也可对故障节点快速隔离定界定位。

  1. 2K+X 节点准备好后进行训前检查，保障测试环境处于健康状态。所有节点完成巡检后，按训练脚本正式启动模型训练测试。
  2. 训练过程报错，找到首报错节点导出首报错节点日志，分析失败原因。同时从 X 节点选备用节点替换报错节点，让训练任务重新拉起。
  3. 完成所有训练迭代后，导出日志进行训练测试结果分析和输出测试报告。

测试环境具体核心步骤

序号	方法	作用	适用场景		
			训前	训中	训后
1	单核压测	识别风险 NPU/GPU 模组	✅		
2	单机压测	首错节点快速识别，在线故障快速诊断	✅		
3	单机训练	首错节点快速识别，在线故障快速诊断	✅		
4	通信压测	速率符合预期，定位慢节点和跨 Leaf 问题	✅		
5	集群训练	拉起训练任务，批量检测模型镜像文件	✅		
6	告警监控	训练前确认，确保过程过程能够发现问题	✅	✅	
7	集群健康检查	二次健康检查，链路闪断、端口 down、防火墙等配置	✅		
8	参数链路检查	定位慢节点和跨 Leaf 问题，GPU/NPU 间无阻塞	✅	✅	✅
9	训练进程监控	进程状态、端口状态、芯片健康状态		✅	
10	残留清理恢复	清理残留进程和容器，预制业务流批量执行	✅		✅

02

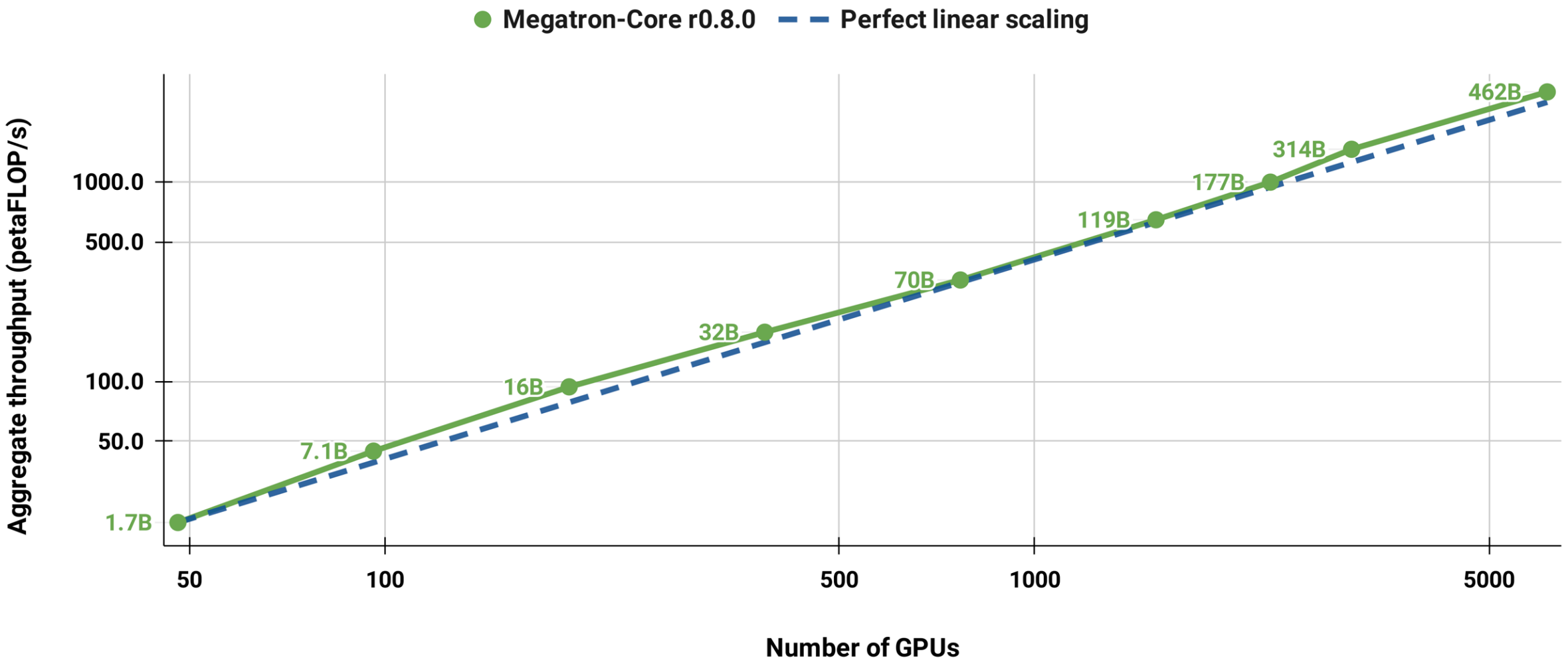
测试方案

英伟达 NVIDIA Megatron 测试方案

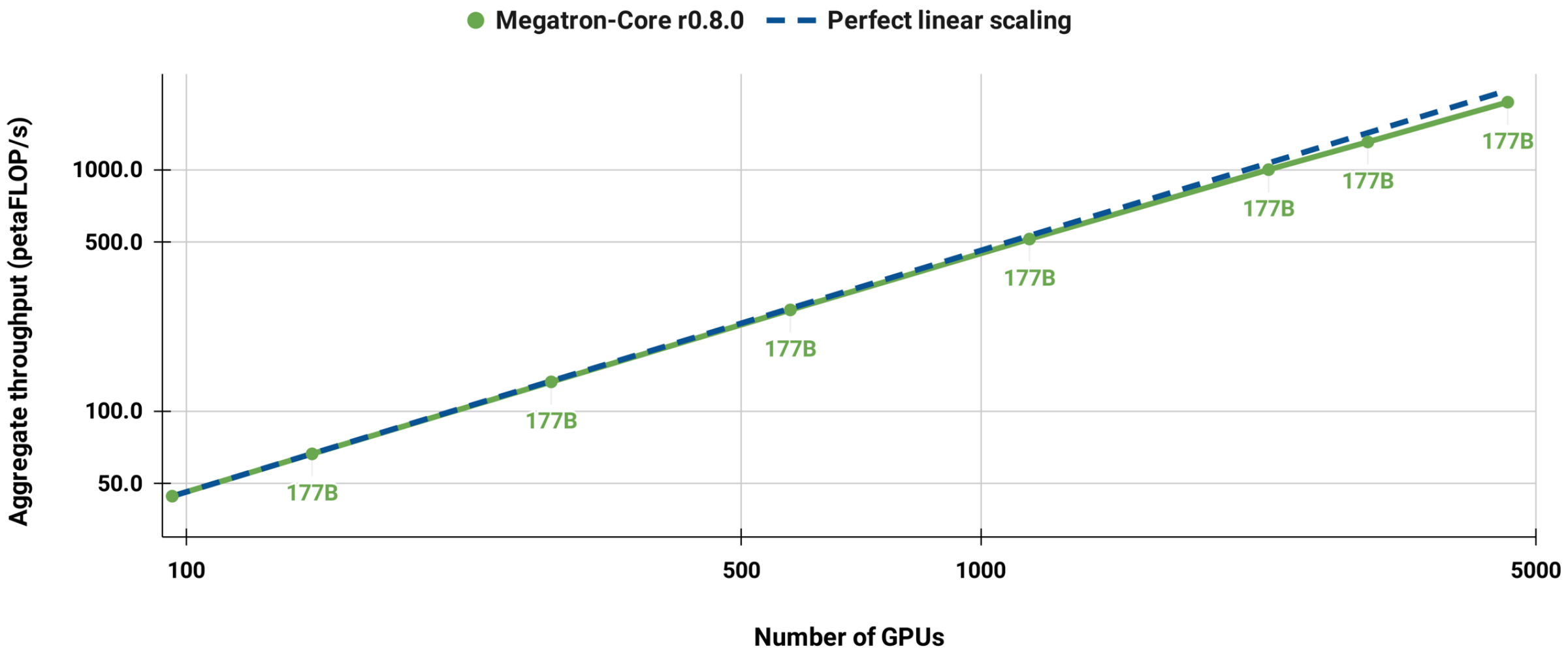
- Our weak scaled results show superlinear scaling (MFU increases from 41% for the smallest model co nsidered to 47-48% for the largest models); this is because larger GEMMs have higher arithmetic inte nsity and are consequently more efficient to execute.

Model size	Attention heads	Hidden size	Number of layers	Tensor MP size	Pipeline MP size	Data-parallel size	Number of GPUs	Batch size	Per-GPU teraFLOP/s	MFU	Aggregate petaFLOP/s
1.7B	16	2048	24	1	1	48	48	192	408.8	41%	19.6
7.1B	32	4096	30	2	1	48	96	192	465.9	47%	44.7
16B	48	6144	32	4	1	48	192	192	489.1	49%	93.9
32B	56	7168	48	8	1	48	384	192	459.6	46%	176.5
70B	64	8192	84	8	2	48	768	384	419.7	42%	322.3
119B	80	10240	92	8	4	48	1536	768	420.5	43%	645.9
177B	96	12288	96	8	6	48	2304	1152	432.8	44%	997.2
314B	128	16384	96	8	8	48	3072	1536	474.4	48%	1457.4
462B	144	18432	112	8	16	48	6144	3072	459.9	47%	2825.6

英伟达 NVIDIA Megatron 测试方案



英伟达 NVIDIA Megatron 测试方案



模型测试方案

1. 模型按照参数量和匹配的计算节点数递增，并提供 Dense 类模型和 MoE 稀疏类模型：

Model size	Attention heads	Hidden size	Number of layers	Tensor MP size	Pipeline MP size	Data-parallel size	Number of GPUs	Batch size	Per-GPU teraFLOP/s	MFU	Aggregate petaFLOP/s
1.7B	16	2048	24	1	1	48	48	192	408.8	41%	19.6
7.1B	32	4096	30	2	1	48	96	192	465.9	47%	44.7
16B	48	6144	32	4	1	48	192	192	489.1	49%	93.9
32B	56	7168	48	8	1	48	384	192	459.6	46%	176.5

2. 根据不同的并行策略、序列长度、GBS 来递增计算节点，测试其他性能指标：

Nodes	NPUs	Parallel					GBS	序长	带宽	1 step			吞吐率	线性度	TFLOPS	MFU	训练时长	MTTR
		TP	PP	DP	EP	CP				单迭代时长	训练时间波动	训练性能波动						

集合通信测试方案

- **通信测试目标：**

Content
评估不同通信算法下通信性能
测试不同数据包大小对通信带宽的影响
评估网络拓扑结构和拥塞情况
验证 AI 集群通信效率
识别通信瓶颈

- **测试维度：**

维度	Content
通信源语	Ring AllReduce, HD AllReduce, All-to-All, Broadcast, Reduce Scatter, AllGather
数据规模	小包 1KB、中包 1~100M、大包 500M~1GB~2GB
拓扑结构	单机多卡、跨节点、跨机架、全集群
通信协议	TCP/IP、RDMA、NVLink、InfiniBand
并行策略	数据并行、模型并行、流水线并行、专家并行

集合通信测试方案

编号	通信算法	节点数	GPU 数	包大小	拓扑结构	平均延迟 (ms)	带宽 (GB/s)	网络利用率 (%)	失败重试数 (次)	网卡阻塞率 (%)
T001	Ring AllReduce	1	8	1MB	Ring					
T002	HD AllReduce	16	128	10MB	Tree + Ring					
T003	AllGather	128	1024	100MB	Fully Connected					
T004	Broadcast	1024	8192	1GB	Tree					
T005	ReduceScatter	1024	8192	1GB	Ring					
T006										
T007										

线性度测试方案

- 在万卡 AI 集群中评估通信的线性度 Scaling Linearity，是为了衡量随着集群规模扩大时，通信效率是否保持理想比例提升或下降。好的通信线性度意味着集群规模增长能够有效扩展，反之说明存在通信瓶颈、网络拥塞、协议限制等问题。

目标	描述
检查通信吞吐随节点/GPU 数量扩大的变化趋势	是否接近线性？是否存在拐点？
分析不同通信算法下的可扩展性差异	Ring vs HD vs Tree 等
定位通信瓶颈	是网络带宽瓶颈？还是软件栈瓶颈？
为大规模训练提供性能预测依据	判断是否值得继续扩展至万卡级别

线性度测试方案

测试规模	消息大小 (MB)	理论带宽 (GB/s)	有效带宽 (GB/s)	单卡带宽衰减 (%)	延迟P99 (ms)	同步开销占比 (%)	线性度 (%)	异常事件
256卡	32	(基准)	183.5	0% (基准)	4.8	8.2%	100%	无
1,024卡	32	=256卡带宽*4	178.9	2.5%	7.1	13.5%	97.5%	端口波动
4,096卡	32	=256卡带宽*16	164.2	10.5%	15.3	24.7%	89.5%	重试0.3%
8,192卡	32	=256卡带宽*32	149.8	18.4%	28.6	36.2%	81.6%	阻塞事件
10,240卡	32	=256卡带宽*40	137.4	25.1%	41.2	47.8%	74.9%	3节点超时

瓶颈现象	解决方案	预期收益
8000 卡以上尾延迟突增	NCCL_IGNORE_CPU_AFFINITY=1	-15% P99 延迟
同步时间占比 >40%	改用 Hybrid Hierarchical All-Reduce	-30% 同步开销
跨柜带宽衰减 >20%	调整 ECMP 路由权重	+18% 有效带宽

总结与思考

问题焦点 + 具体措施 + 量化目标

1. 集群可靠性压力测试，充分挖掘硬件故障
2. 长周期连续训练稳定性测试
3. 分布式训练线性度与扩展性测试
4. 自动化监控与告警体系建设
5. 训练任务失败归因分析机制建立

Summary：跟 AI 集群模型，跟硬件磨合，跟基础软件磨合

1 问题焦点 + 具体措施 + 量化目标

- **集群可靠性压力测试，充分挖掘硬件故障**
- **目标：**大部分硬件问题均是早期失效导致，暴露早期硬件缺陷
  - 交接前连续运行不同参数规模、不同类型的等模型任务（10B~100B参数），覆盖 90% 计算柜
  - 设置不同 BS、精度（FP16、BF16、TF32）、分布式策略（DDP、FSDP、ZeRO）进行测试
  - 收集单步耗时、吞吐量、GPU利用率、通信带宽、内存占用不同指标
  - **量化指标**：达成 ≥200 小时连续训练零断点故障，硬件失效率压降至 <0.5%/周

2 问题焦点 + 具体措施 + 量化目标

- **长周期连续训练稳定性测试**
- **目的：** 验证 AI 集群在长时间高负载下的稳定性和容错能力
  1. 选择典型的大模型，进行连续 200h 以上的训练任务（e.g. DeepSeek/Qwen）
  2. 设置检查点保存机制 checkpoint，定期保存模型状态
  3. 模拟故障注入，观察任务是否能自动恢复、恢复时长、性能（e.g. 断网、重启某节点、GPU掉卡）
  4. 记录日志、监控数据（e.g. GPU/NPU 温度、风扇转速、电源状态）

3 问题焦点 + 具体措施 + 量化目标

- **分布式训练线性度与扩展性测试**
- **目的：** 验证随着节点数增加，训练性能是否呈近似线性提升
  - 使用相同模型在不同节点数量下执行训练和推理任务（e.g. 4、8、16、32、64、128 节点）
  - 比较步耗时、吞吐量、TFLOPS 利用率、集合通信效率
  - 分析是否存在通信瓶颈、是否有明显的非线性下降、是否出现计算空闲等待现象
  - **量化指标：** 扩展性曲线图、通信开销占比分析、最优并行策略推荐

4 问题焦点 + 具体措施 + 量化目标

- **自动化监控与告警体系建设**
- **目的：**构建完整的集群健康监控体系，提前发现潜在的集群问题
  - **设置监控维度：**GPU/NPU 利用率、显存使用、温度、功耗；网络延迟、带宽、丢包率；存储IO吞吐、响应延迟；任务状态、 Pod 状态、调度延迟等
  - **设置阈值告警规则：**实现 GPU 故障预警、网络拥塞预警、资源争抢预警

5 问题焦点 + 具体措施 + 量化目标

- **训练任务失败归因分析机制建立**
- **目的：** 为后续快速定位训练失败原因提供依据
  - 建立分类标签体系：硬件类（e.g. GPU卡死、内存溢出、PCIe通信异常）、网络类（e.g. 通信超时、NCCL错误）、软件类（e.g. PyTorch Bug、CUDA OOM）、配置类（e.g. 参数设置不合理）
  - 对训练任务失败进行归类记录，无论是代码错误、硬件故障还是资源不足
  - 结合日志、堆栈信息、监控数据进行根因分析，提炼白皮书和行业标准

Question?

- 总结？干就完了！摸着石头过河，总结经验，下一个视频更精彩！

Thank you

把 AIInfra 带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI Infra to every person, home and organization for a fully connected, intelligent world.

ZOMI

GitHub github.com/Infrasys-AI/AIInfra

Book infrasys-ai.github.io

引用与参考

1. http://www.ccw.com.cn/ai/2022-03-01/23727.html
2. https://cloud.tencent.cn/developer/article/1705673

PPT 开源在： https://github.com/Infrasys-AI/AIInfra