Workload	Container Orchestrator	Distributed compute engine	Training & inference framework	Outcomes
LLM training & evaluation Fine-tuning 7B to 70B models on A100s & H100s; model evals	**Kubernetes**-based Michelangelo Job Controller across multiple AZs (for GPU availability and scale)	**Ray** for coordinating and scaling	**PyTorch DDP**, **DeepSpeed**, Hugging Face Transformers, **vLLM** for offline scoring	GPU memory reduction enabling 2-7x larger batches leading to 2-3x throughput increase on Llama-2 70B. ✧ (blog, blog)
Deep-learning & classical ML training Distributed training, hyperparameter optimization	Originally Peloton, now **Kubernetes**-based Michelangelo Job Controller	Originally Spark, now **Ray**	Horovod, XGBoost, **PyTorch**	Improved scalability and reliability. ✧ (blog, blog, blog)
Batch inference Classical models and LLMs	Originally Peloton, now **Kubernetes**-based Michelangelo Job Controller	Spark, **Ray**	TensorFlow, **PyTorch**, XGBoost, **vLLM**, Triton for embeddings	Scalability and GPU support. Multi-GPU inference. ✧ (blog, talk)
Model serving Latency sensitive	Originally Peloton, now **Kubernetes**-based Michelangelo Job Controller	Michelangelo online prediction service	TensorFlow, **PyTorch**, previously served with Neuropod, now Triton	Framework agnostic, support for low-latency GPU serving. ✧ (blog)
Marketplace-incentive optimization Adjusting incentives & discounts across thousands of cities	Originally Peloton, now **Kubernetes**-based Michelangelo Job Controller	Originally Spark, switched to Spark + **Ray** hybrid	CVXOPT and pure Python logic for optimization	40x overall speed-up. Reduced job deployment from 15–20 min to 2 min. Improved iteration speed. ✧ (blog)