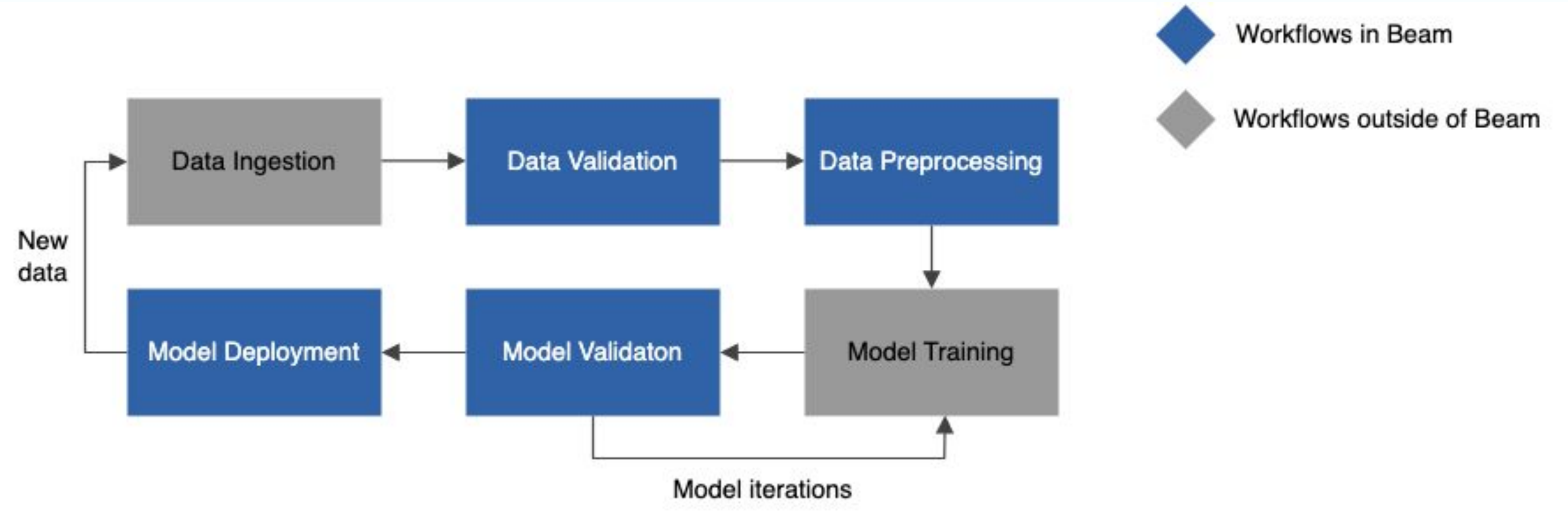
How Beam Serves Models with vLLM

Danny McCormick

BEAM SUMMIT
NYC 2025

The ML lifecycle



Inference with Beam

Challenges of Distributed Inference

- Efficiently loading models
- Batching
- Model Updates
- Using multiple models

Distributed Inference with Beam

- Beam takes care of all of this with the RunInference transform
- Loads model, batches inputs, handles updates, and plugs into DAG

```
RunInference(model_handler=<config>)
```

RunInference

```
>>> data = numpy.array([10, 40, 60, 90],
...                           dtype=numpy.float32).reshape(-1, 1)

>>> model_handler = PytorchModelHandlerTensor(
...    model_class=LinearRegression,
...    model_params={'input_dim': 1, 'output_dim': 1},
...    state_dict_path='gs://path/to/model.pt')

>>> with beam.Pipeline() as p:
...   predictions = (
...       p
...       | beam.Create(data)
...       | beam.Map(torch.Tensor) # Map np array to Tensor
...       | RunInference(model_handler=model_handler)
...       | beam.Map(print))
```

Basic Inference Demo

colab.sandbox.google.com/github/apache/beam/blob/master/examples/notebooks/beam-ml/run_inference_huggingface.ipynb

What is vLLM?

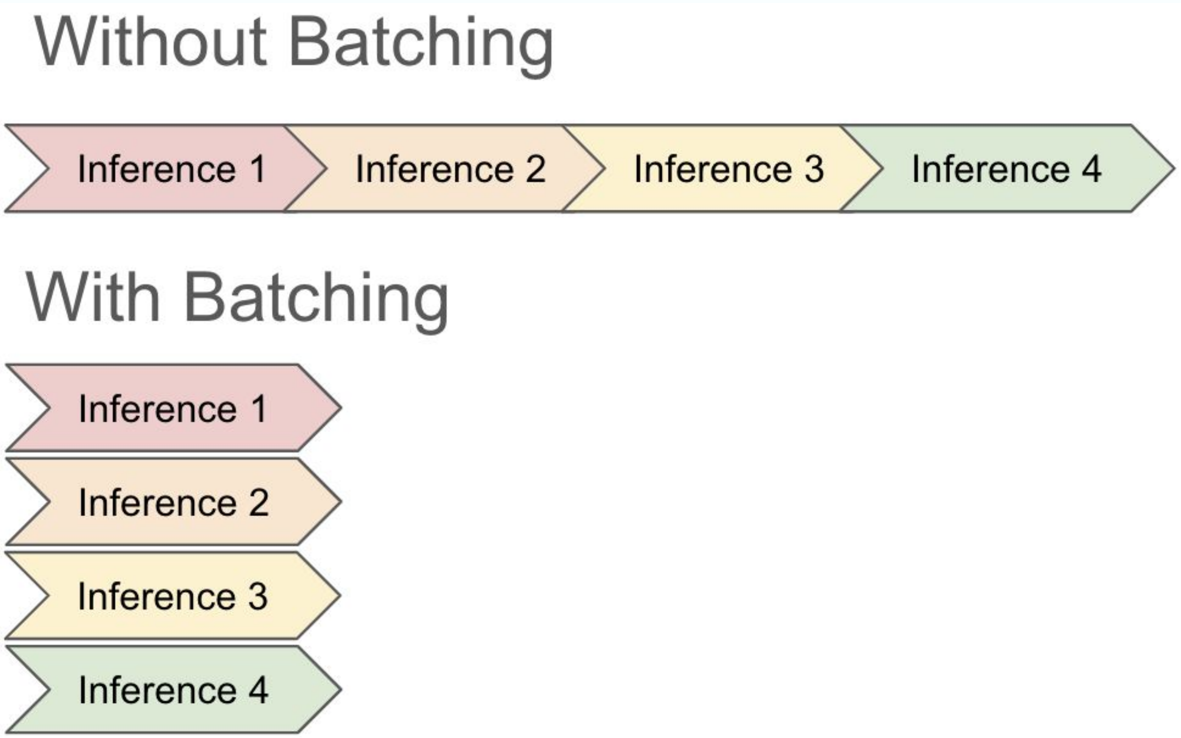
BEAM SUMMIT NYC 2025

What is vLLM?

- Open source library for serving large language models
- Takes advantage of the specific architectures of LLMs to perform optimizations a generalized framework can't
    - Examples: continuous batching, PagedAttention

Traditional Batching

- ML frameworks operate more efficiently when running multiple records in parallel

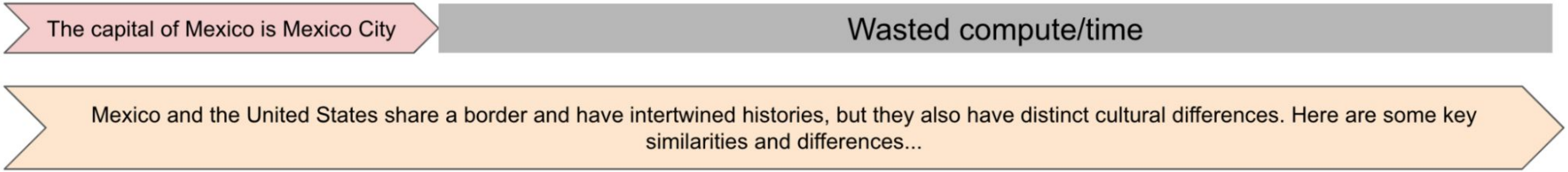


What if some records take longer than others?

- If some records finish early, they have to wait for the others

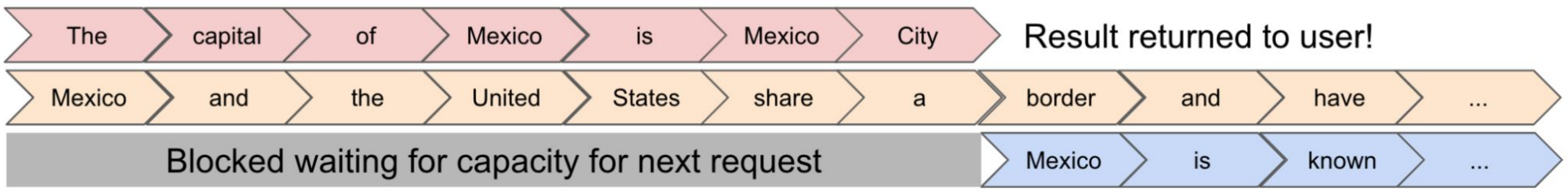
LLMs almost always run into this

- LLMs basically perform an inference (or chain of inferences) per token
- The longer the input, the longer each inference takes (and often the more tokens that need to be generated)



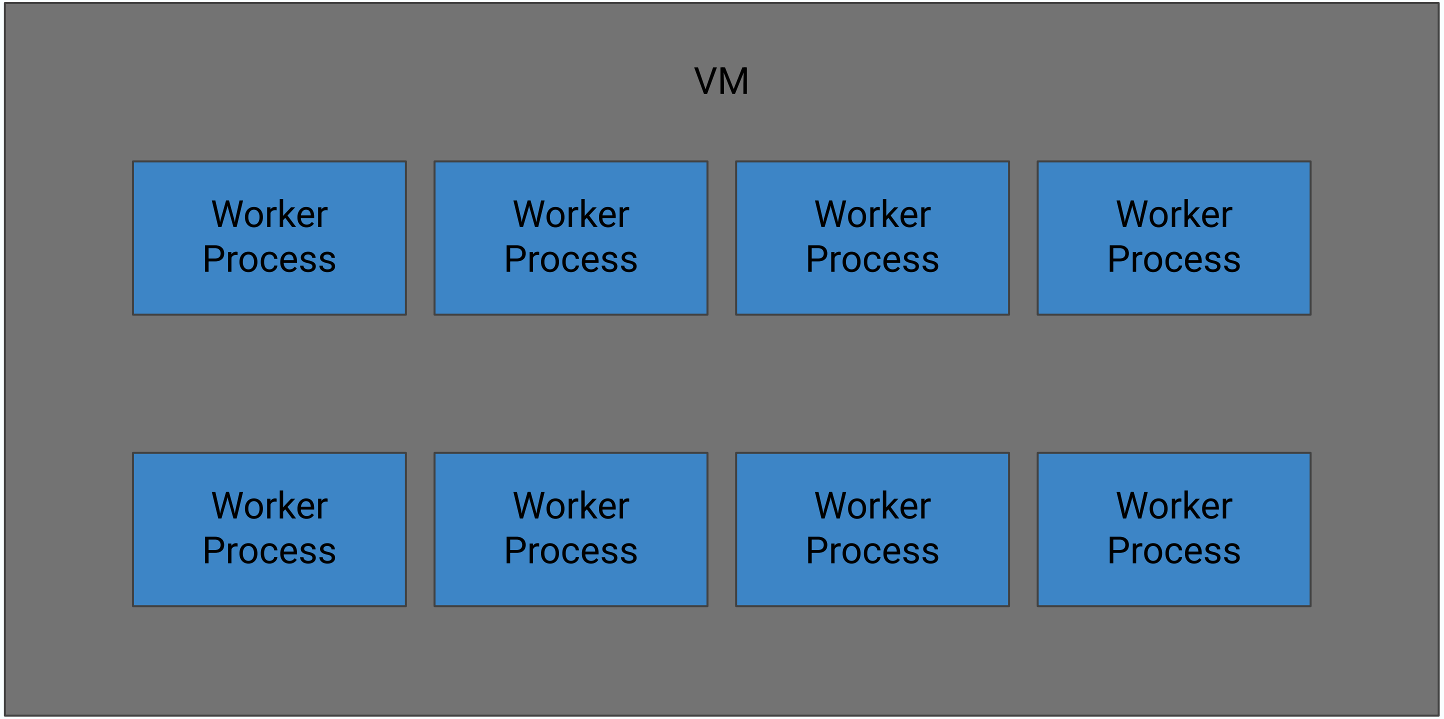
A better way: continuous batching!

- Batch at the token level instead of the record level

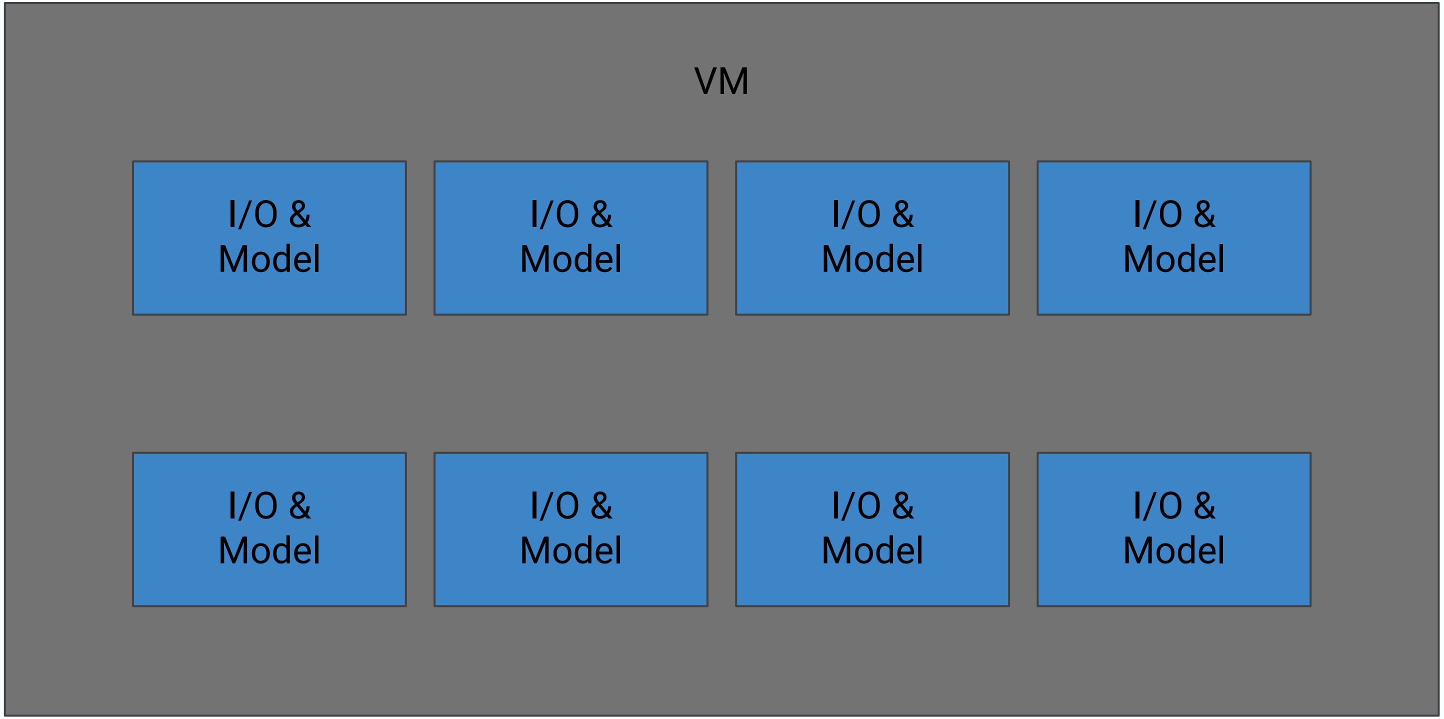


Back to Beam - how Beam does do Model Management

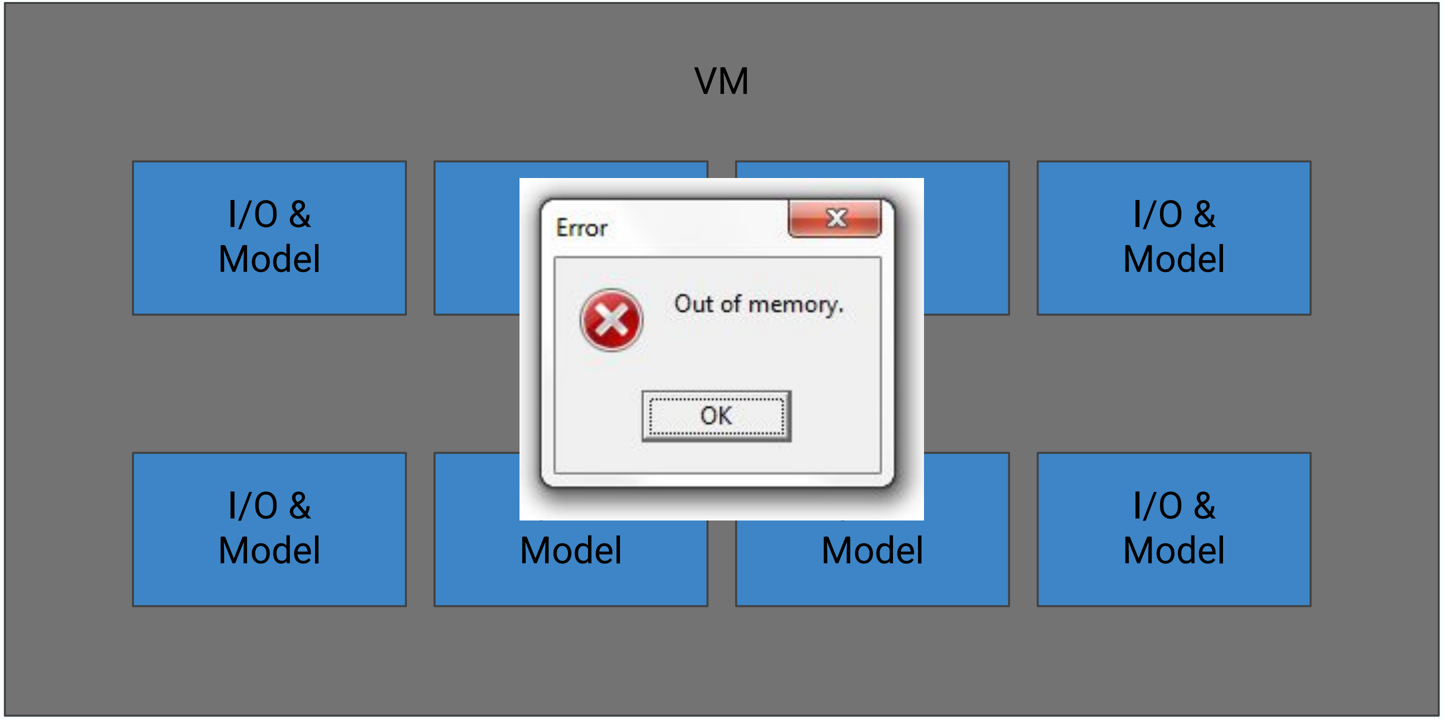
Distributed Runner Architecture



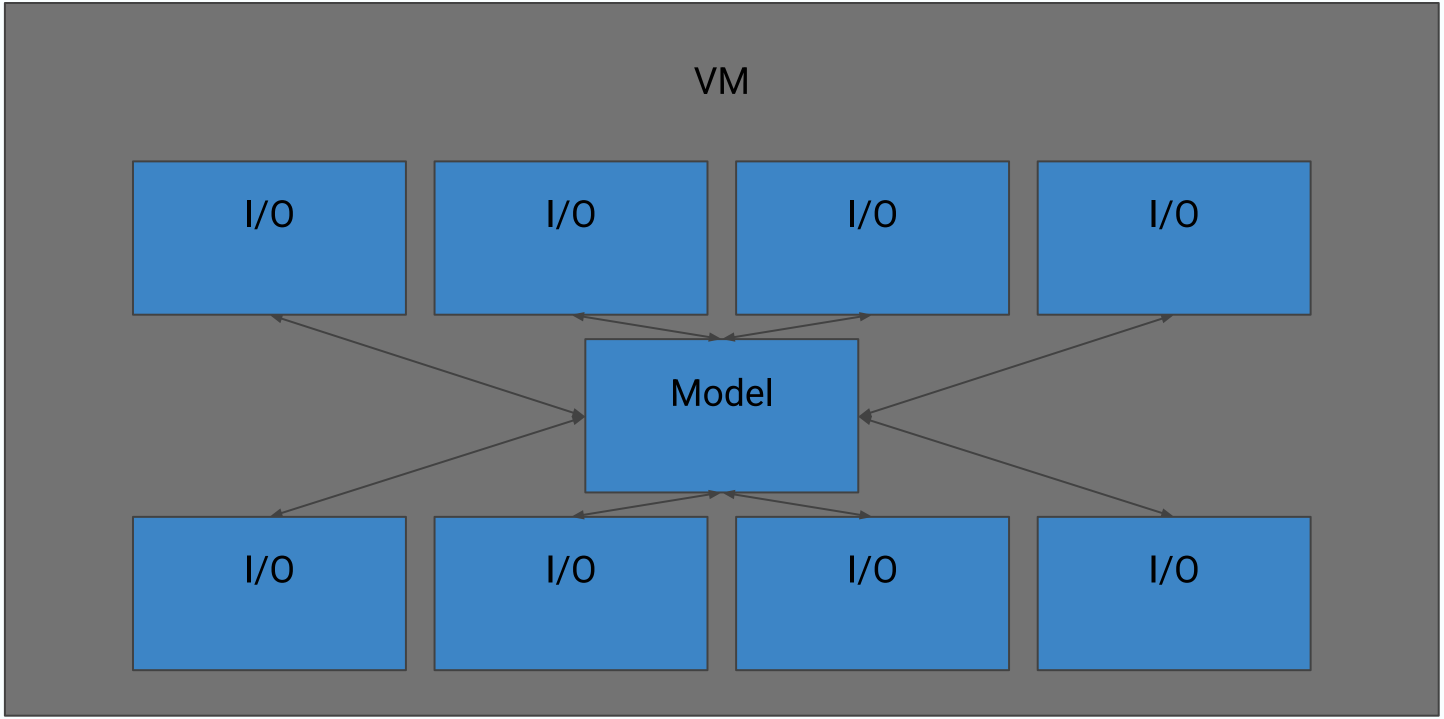
Works well for small models



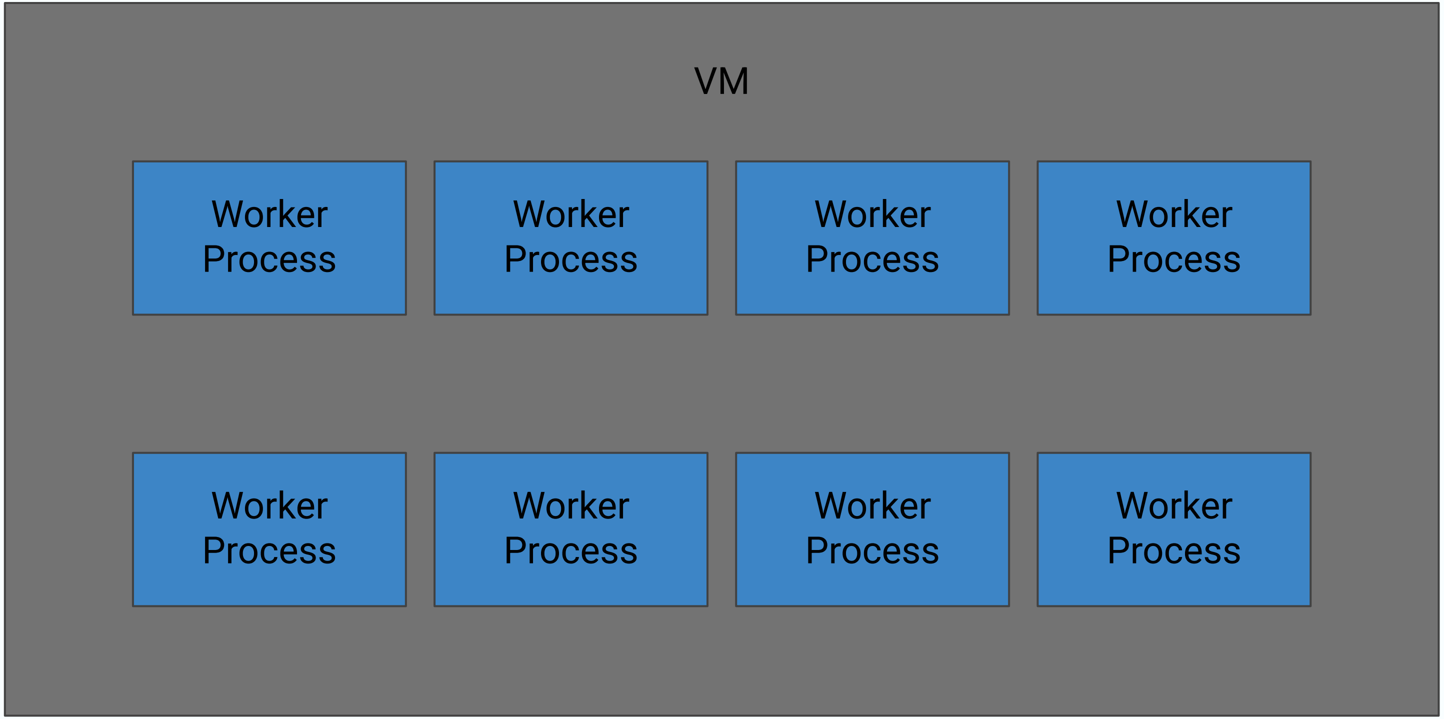
But poorly for big models



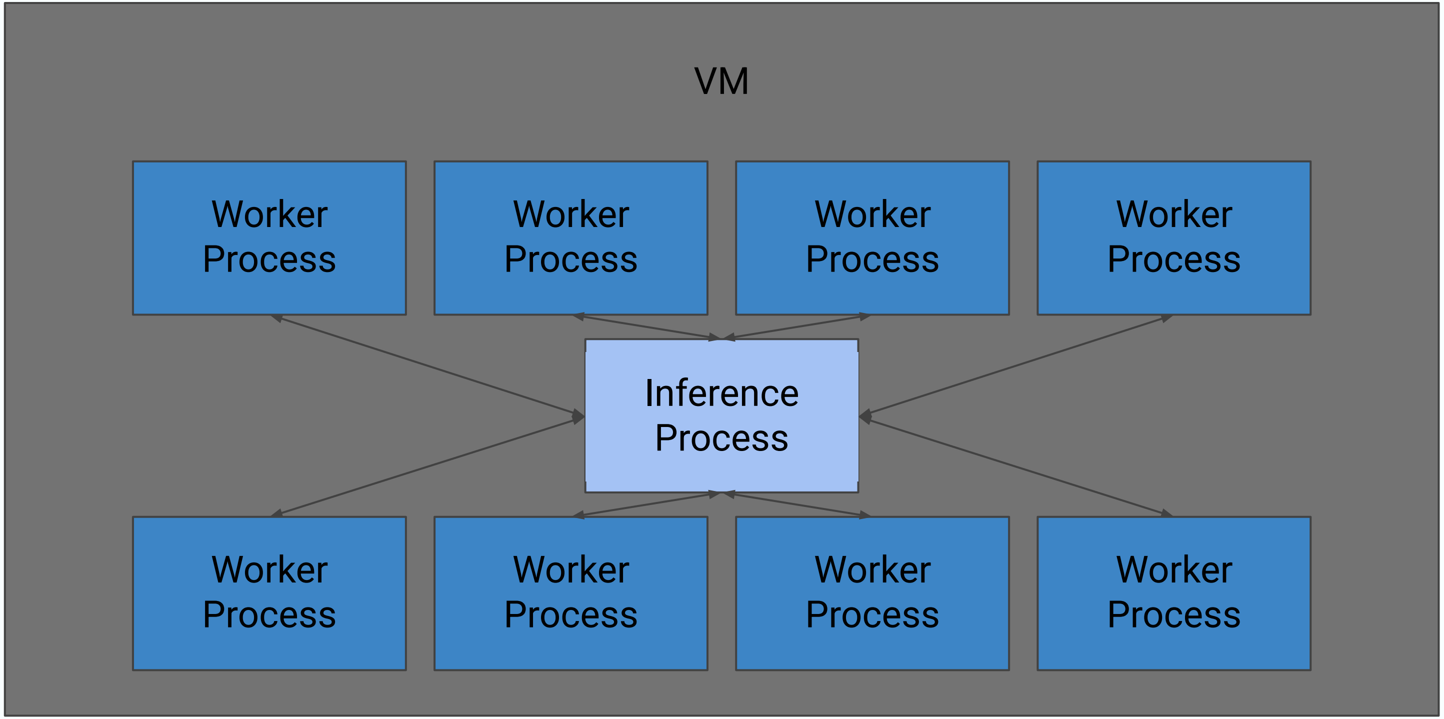
Ideal Large Model Configuration



How do we map ideal model configuration to this?

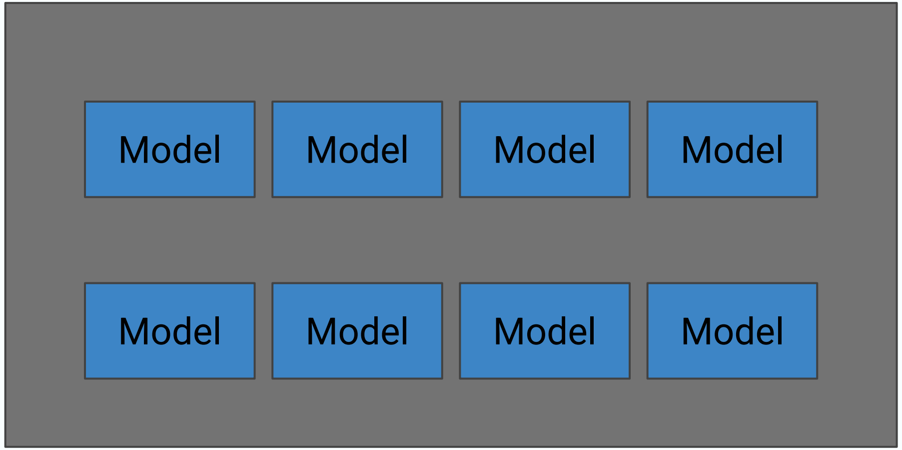


Automatically spin up a dedicated process!



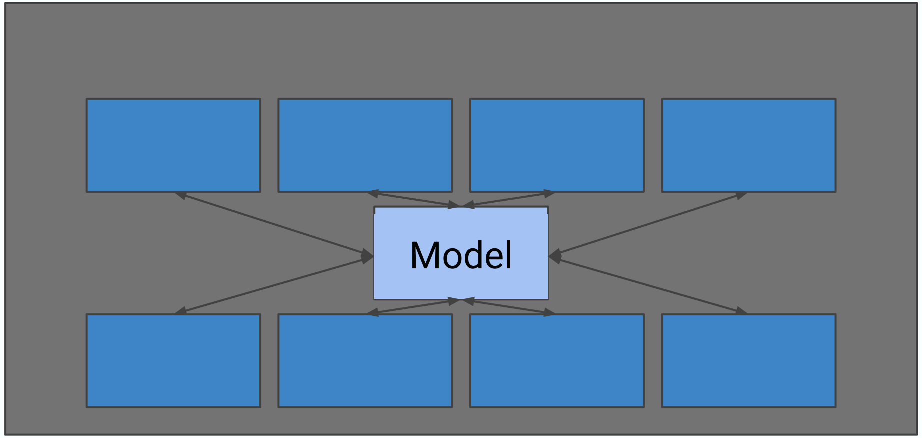
Configuration for a small model

```
>>> model_handler = PytorchModelHandlerTensor(
...     model_class=LinearRegression,
...     model_params={'input_dim': 1, 'output_dim': 1},
...     state_dict_path='gs://path/to/model.pt')

>>> pcoll | RunInference(model_handler=model_handler)
```



Configuration for a small model

```
>>> model_handler = PytorchModelHandlerTensor(
...     model_class=LinearRegression,
...     large_model=True,
...     model_params={'input_dim': 1, 'output_dim': 1},
...     state_dict_path='gs://path/to/model.pt')

>>> pcoll | RunInference(model_handler=model_handler)
```

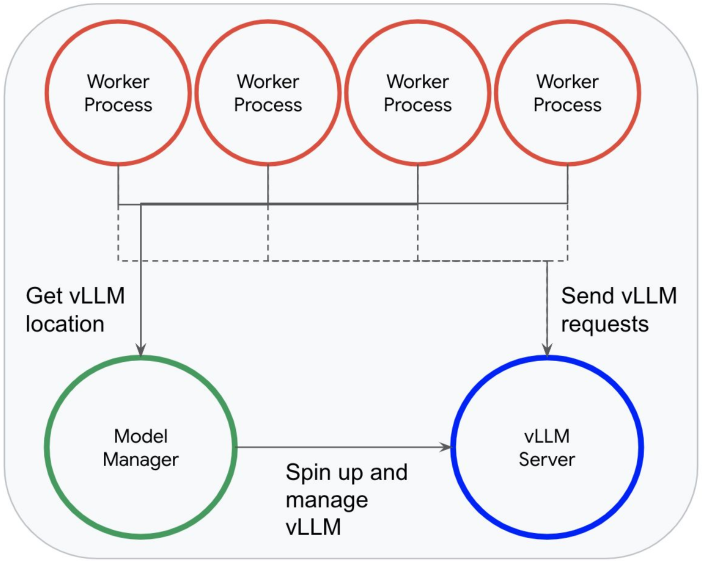


But how does vLLM fit in

vLLM isn't just a model, it is a standalone service

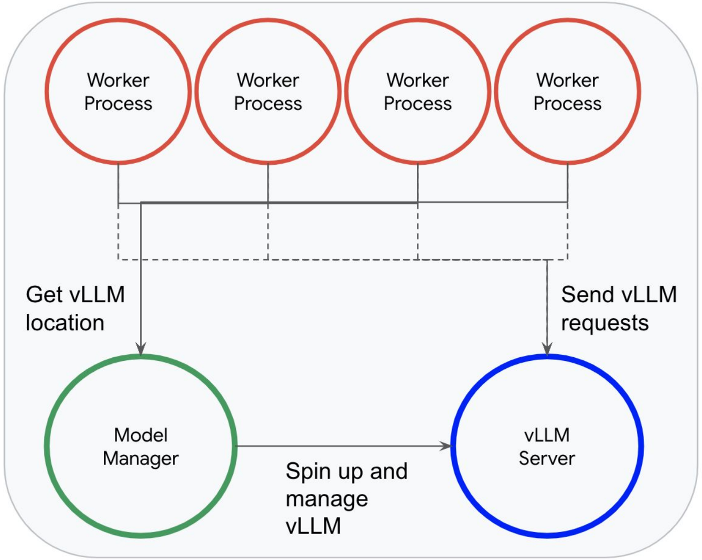
vLLM

Introducing the model manager!



Introducing the model manager!

- Model Manager (Inference process) now spins up a reference to vLLM server
- Workers talk to vLLM server directly
- Model manager manages vLLM lifecycle
  - Start up
  - Dealing with stuckness/failures
  - Teardown



Usage

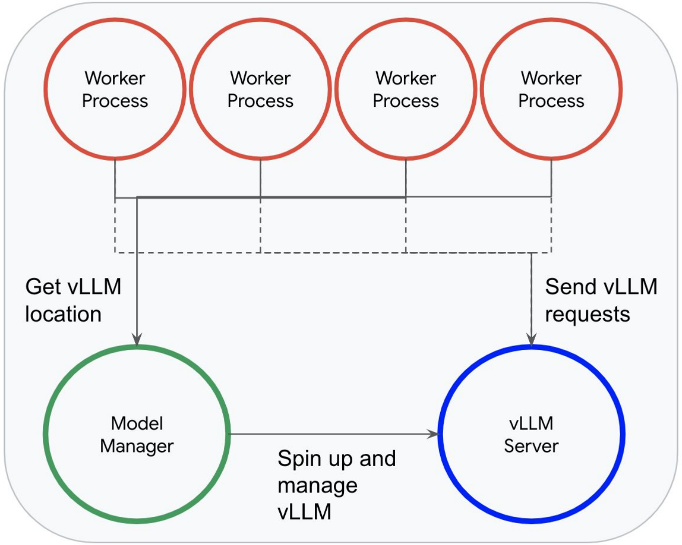
```
>>> prompts = ["One cause of the console being blank is", "If
you're experiencing network issues","If the button isn't working",
"If you can't submit your job"]

>>> mh = VLLMCompletionsModelHandler('my_favorite_llm')

>>> with beam.Pipeline() as p:
...   predictions = (
...       p
...       | beam.Create(prompts)
...       | RunInference(model_handler=mh)
...       | beam.Map(print))
```

vLLM demo

colab.sandbox.google.com/github/apache/beam/blob/master/examples/notebooks/beam-ml/run_inference_vllm.ipynb

Performance

- Varies greatly model to model
- With one example pipeline using Google's Gemma 2b model, saw a 23x reduction in number of CPU/GPU core hours when switching from pure Pytorch to vLLM



Danny Mccormick

QUESTIONS?

github.com/damccorm
linkedin.com/in/danny-mccormick-a044b1103

BEAM SUMMIT
NYC 2025