Splitwise: Efficient Generative LLM Inference Using Phase Splitting

Each LLM inference request has two distinct phases with different resource requirements:

### Prompt computation and token generation phases

- **User prompt** processed in parallel
- **First token** transfer
- **Rest of the output tokens** transfer
- **Memory intensive** (relies on KV cache)
- **Limited batching benefits**
- **Batching improves throughput**

### Example 1: Batching effects

- **Prompt phase** throughput:
  - **90%** of requests
  - **1100 ms**
- **Token phase** throughput:
  - **200 ms**

### Example 2: Power usage

- **Prompt phase** power usage:
  - **1x**
- **Token phase** power usage:
  - **1.1x**

Splitwise splits inference across different servers to enable phase-specific resource management:

1. **Transfer request state over P2P GPU Infiniband; optimize with parallel and overlapped transfers**

   - **Prompt server**
     - User prompt
     - First token
     - Prompt GPUs
     - KV cache transfer
   
   - **Token server**
     - Rest of the output tokens
     - Token GPUs

2. **Split cluster into three server pools and use phase-specific resource management at scale**

   - **Prompts pool**
     - Small batches
     - Full power
   
   - **Mixed pool**
     - Large batches
     - Power capped
   
   - **Token pool**
     - Compute-intensive GPUs

Splitwise clusters are much more resource efficient than existing clusters:

**Result 1:** Splitwise transfers request state with less than ~0.8% end-to-end overhead on average

**Result 2:** Clusters designed using Splitwise provide much higher throughput than existing clusters

<table>
<thead>
<tr>
<th>Throughput</th>
<th>Baseline</th>
<th>Splitwise homogeneous</th>
<th>Splitwise heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Servers</td>
<td>1x</td>
<td>1x</td>
<td>0.73x</td>
</tr>
<tr>
<td>Cost</td>
<td>1x</td>
<td>1x</td>
<td>1.14x</td>
</tr>
<tr>
<td>Power</td>
<td>Throughput</td>
<td>2.4x</td>
<td>2.6x</td>
</tr>
</tbody>
</table>

Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, Ricardo Bianchini

Paper, code, traces at aka.ms/splitwise