Lab 6: Optimization and Scaling Techniques for Large Language Models on Modern Cloud Infrastructure with GPU Accelerators

Mario Ventura

0. Hugging Face Transformers

Simplificar entrenamiento y despliegue modelos avanzados y facilitar acceso a LLMs.

Características clave:

- Librería abierta (open) para PLN, visión y voz.
- Multitud de modelos (Model Hub)
- Soporte PT & TF
- APIs incluidas

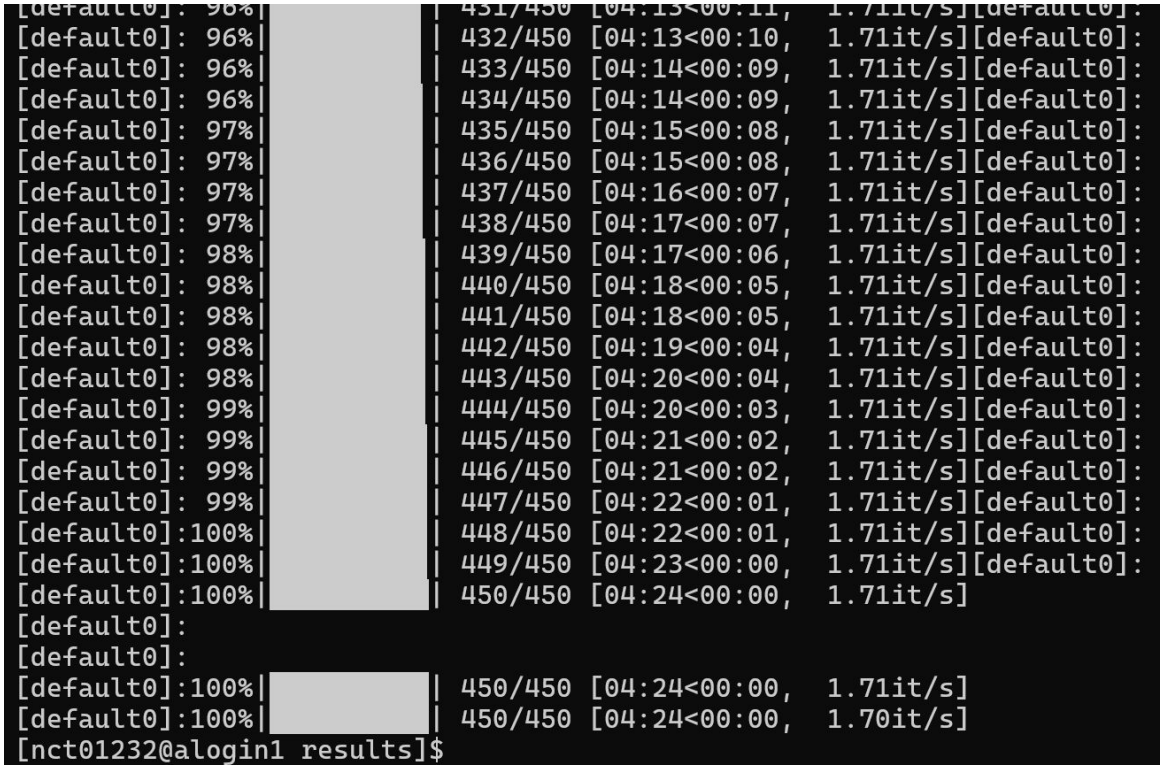
1. Baseline Experiment

Ejecutar script **submit.LLM.task.1.slurm** con:

- Job ID: 17261309 (se asigna automáticamente)
- GPUs: **1**
- Batch Size: **6**

Objetivo: Establecer referencia para comparar optimizaciones futuras

1. Baseline Experiment



.err

1. Baseline Experiment

Resultados en el archivo **R-LLM_task1.*JobID*.out**

```
[nct01232@alogin1 results]$ cat R-LLM_task1.17261309.out
START TIME: Wed Mar 19 18:51:51 CET 2025
ACCELERATE_MIXED_PRECISION=no torchrun     --nproc_per_node 1     --nnodes 1     --node_rank $SLURM_PROCID     --rdzv_endpoint as03r2b31:6000     --rdzv_backend c10d     --max_restarts 0     --tee 3
     ./benchmark.py --path_to_model Llama-3.2-1B     --run_name NODES-1-GPUs-1-LLAMA3.2-1B-MODEL-PRECISION-fp32-MIXED-PRECISION-no-ATTN-eager-adamw_torch-TC-false-LIGER-false-SEQLEN-1024-MBS-mak6-7b0
4c241-9b9b-4121-8278-8630d113f06b     --max_steps 450     --sequence_length 1024     --per_device_train_batch_size 6     --model_precision fp32     --attn eager     --torch_compile false     --use_li
ger_kernel false     --optim adamw_torch     --output_dir ./results/output     --save_strategy no     --report_to none

Current hostname: as03r2b31
[default0]:[2025-03-19 17:52:08,803] [INFO] [real_accelerator.py:203:get_accelerator] Setting ds_accelerator to cuda (auto detect)
[default0]:[03/19/2025 05:52:08 PM] INFO - x86_64-linux-gnu-gcc -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O2 -Wall -g -fstack-protector-strong -Wformat -Werror=format-security -g -fwrapv
 -O2 -fPIC -c /scratch/tmp/17261309/tmpu3ck2a6q/test.c -o /scratch/tmp/17261309/tmpu3ck2a6q/test.o
[default0]:[03/19/2025 05:52:09 PM] INFO - x86_64-linux-gnu-gcc /scratch/tmp/17261309/tmpu3ck2a6q/test.o -laio -o /scratch/tmp/17261309/tmpu3ck2a6q/a.out
[default0]:[03/19/2025 05:52:09 PM] INFO - x86_64-linux-gnu-gcc -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O2 -Wall -g -fstack-protector-strong -Wformat -Werror=format-security -g -fwrapv
 -O2 -fPIC -c /scratch/tmp/17261309/tmp5bu3rbtw/test.c -o /scratch/tmp/17261309/tmp5bu3rbtw/test.o
[default0]:[03/19/2025 05:52:09 PM] INFO - x86_64-linux-gnu-gcc /scratch/tmp/17261309/tmp5bu3rbtw/test.o -L/usr/local/cuda -L/usr/local/cuda/lib64 -lcufile -o /scratch/tmp/17261309/tmp5bu3rbtw/a.out
[default0]:[03/19/2025 05:52:09 PM] INFO - x86_64-linux-gnu-gcc -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O2 -Wall -g -fstack-protector-strong -Wformat -Werror=format-security -g -fwrapv
 -O2 -fPIC -c /scratch/tmp/17261309/tmp34yzm0t0/test.c -o /scratch/tmp/17261309/tmp34yzm0t0/test.o
[default0]:[03/19/2025 05:52:09 PM] INFO - x86_64-linux-gnu-gcc /scratch/tmp/17261309/tmp34yzm0t0/test.o -laio -o /scratch/tmp/17261309/tmp34yzm0t0/a.out
[default0]:{'train_runtime': 264.2184, 'train_samples_per_second': 10.219, 'train_steps_per_second': 1.703, 'train_loss': 6.954143337673611, 'epoch': 0.27}
END TIME: Wed Mar 19 18:56:36 CET 2025
[nct01232@alogin1 results]$
```

TASK	Job ID	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger
1	17261309	1	6	10.465	20	fp32	NO	eager	FALSE

1. Baseline Experiment

Conclusiones:

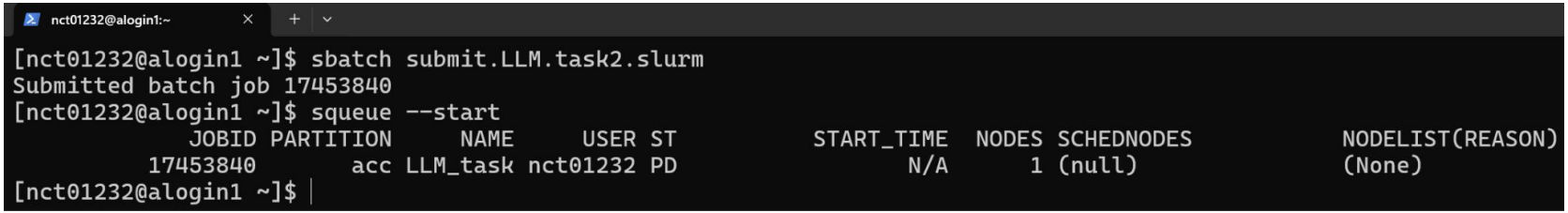
- **Throughput:** El modelo procesa **10,465 tokens por segundo** en un solo GPU con un batch size pequeño (6)
- **Memoria reservada: 20 GiB**. Es moderada, considerando que el H100 tiene 64 GiB de VRAM.

Falta de optimizaciones y batch size pequeño limitan el throughput al no maximizar el uso de Tensor Cores.

2. Finding the Out-Of-Memory Limit

Ejecutar script **submit.LLM.task.2.slurm** con:

- Job ID: 17453840 (se asigna automáticamente)
- GPUs: **1**
- Batch Size: **7**



2. Finding the Out-Of-Memory Limit

Out Of Memory!

```
[default0]:[rank0]:     return forward_call(*args, **kwargs)
[default0]:[rank0]:   File "/usr/local/lib/python3.10/dist-packages/transformers/models/llama/modeling_llama.py", line 1214, in forward
[default0]:[rank0]:     loss = self.loss_function(logits=logits, labels=labels, vocab_size=self.config.vocab_size, **loss_kwargs)
[default0]:[rank0]:   File "/usr/local/lib/python3.10/dist-packages/transformers/loss/loss_utils.py", line 38, in ForCausalLMLoss
[default0]:[rank0]:     shift_logits = logits[..., :-1, :].contiguous()
[default0]:[rank0]: torch.OutOfMemoryError: CUDA out of memory. Tried to allocate 3.42 GiB. GPU 0 has a total capacity of 63.43 GiB of which 1.92 GiB is free. Including non-PyTorch memory, thi
ss has 61.50 GiB memory in use. Of the allocated memory 59.26 GiB is allocated by PyTorch, and 1.17 GiB is reserved by PyTorch but unallocated. If reserved but unallocated memory is large try
 PYTORCH_CUDA_ALLOC_CONF=expandable_segments:True to avoid fragmentation.  See documentation for Memory Management  (https://pytorch.org/docs/stable/notes/cuda.html#environment-variables)
[default0]:
[default0]:  0%|          | 1/450 [00:02<15:26,  2.06s/it]
E0321 16:45:59.103000 140335632058176 torch/distributed/elastic/multiprocessing/api.py:832] failed (exitcode: 1) local_rank: 0 (pid: 196481) of binary: /usr/bin/python
Traceback (most recent call last):
```

El máximo posible con esta configuración es **MICRO_BATCH_SIZE = 6**

3. Mixed Precision

Objetivo: Habilitar **Mixed Precision** para reducir uso de memoria y poder permitir **mayores BATCH SIZE sin causar OOM**.

Ejecución de **submit.LLM.task3.slurm**

```
nct01232@alogin1:~
[nct01232@alogin1 ~]$ sbatch submit.LLM.task3.slurm
Submitted batch job 17455580
```

```
[nct01232@alogin1 results]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          17455580       acc LLM_task nct01232  R       2:07      1 as01r5b04
[nct01232@alogin1 results]$
```

3. Mixed Precision

R-LLM_task3.17455580.out (mejora con bf16)

```
[nct01232@alogin1 results]$ cat R-LLM_task3.17455580.out
START TIME: Fri Mar 21 18:02:26 CET 2025
ACCELERATE_MIXED_PRECISION=bf16 torchrun     --nproc_per_node 1     --nnodes 1     --node_rank $SLURM_PROCID     --rdzv_endpoint as01r5b04:6000     --rdzv_backend c10d     --max_restarts 0     --tee
3      ./benchmark.py --path_to_model Llama-3.2-1B     --run_name NODES-1-GPUs-1-LLAMA3.2-1B-MODEL-PRECISION-fp32-MIXED-PRECISION-bf16-ATTN-eager-adamw_torch-TC-false-LIGER-false-SEQLEN-1024-MBS-mak5
-ca584221-037a-4a4d-8286-f66ee87a848c     --max_steps 450     --sequence_length 1024     --per_device_train_batch_size 5     --model_precision fp32     --attn eager     --torch_compile false     --us
e_liger_kernel false     --optim adamw_torch     --output_dir ./results/output     --save_strategy no     --report_to none
Current hostname: as01r5b04
[default0]:[2025-03-21 17:02:45,896] [INFO] [real_accelerator.py:203:get_accelerator] Setting ds_accelerator to cuda (auto detect)
[default0]:[03/21/2025 05:02:45 PM] INFO - x86_64-linux-gnu-gcc -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O2 -Wall -g -fstack-protector-strong -Wformat -Werror=format-security -g -fwrapv
 -O2 -fPIC -c /scratch/tmp/17455580/tmpv20o2ws8/test.c -o /scratch/tmp/17455580/tmpv20o2ws8/test.o
[default0]:[03/21/2025 05:02:46 PM] INFO - x86_64-linux-gnu-gcc /scratch/tmp/17455580/tmpv20o2ws8/test.o -laio -o /scratch/tmp/17455580/tmpv20o2ws8/a.out
[default0]:[03/21/2025 05:02:46 PM] INFO - x86_64-linux-gnu-gcc -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O2 -Wall -g -fstack-protector-strong -Wformat -Werror=format-security -g -fwrapv
 -O2 -fPIC -c /scratch/tmp/17455580/tmpmw64167u/test.c -o /scratch/tmp/17455580/tmpmw64167u/test.o
[default0]:[03/21/2025 05:02:46 PM] INFO - x86_64-linux-gnu-gcc /scratch/tmp/17455580/tmpmw64167u/test.o -L/usr/local/cuda -L/usr/local/cuda/lib64 -lcufile -o /scratch/tmp/17455580/tmpmw64167u/a.out
[default0]:[03/21/2025 05:02:46 PM] INFO - x86_64-linux-gnu-gcc -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O2 -Wall -g -fstack-protector-strong -Wformat -Werror=format-security -g -fwrapv
 -O2 -fPIC -c /scratch/tmp/17455580/tmpp_azp_p4/test.c -o /scratch/tmp/17455580/tmpp_azp_p4/test.o
[default0]:[03/21/2025 05:02:46 PM] INFO - x86_64-linux-gnu-gcc /scratch/tmp/17455580/tmpp_azp_p4/test.o -laio -o /scratch/tmp/17455580/tmpp_azp_p4/a.out
[default0]:{'train_runtime': 205.7004, 'train_samples_per_second': 10.938, 'train_steps_per_second': 2.188, 'train_loss': 6.961687825520833, 'epoch': 0.23}
[default0]:[03/21/2025 05:06:13 PM] INFO - > Training throughput:     11200.76 Tokens/s/GPU
[default0]:[03/21/2025 05:06:13 PM] INFO - > Max reserved GPU Memory: 58.27
END TIME: Fri Mar 21 18:06:15 CET 2025
[nct01232@alogin1 results]$
```

TASK	Job ID	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger	
1	17261309	1	6	10.465	20	fp32	NO	eager	FALSE	Baseline
2	17454556	1	7	N/A	61'50	fp32	NO	eager	FALSE	OOM-Límite
3	17455580	1	5	11.200'76	58'27	fp32	bf16	eager	FALSE	Mixed Precision

3. Mixed Precision

R-LLM_task3.17455580.out

- **Mixed Precision** (bf16) mejoró el Training Throughput en un 7% respecto al baseline, a pesar de usar un batch size menor (5 vs 6).
- **Max Reserved GPU Memory** se redujo respecto al límite OOM pero sigue siendo alto debido a fp32.

Posiblemente se puede mejorar más con un BATCH SIZE mayor

4. Model Precision

Ejecución de **submit.LLM.task4.slurm**

```
[nct01232@alogin1 ~]$ sbatch  submit.LLM.task4.slurm
Submitted batch job 17457878
[nct01232@alogin1 ~]$ squeue --start
             JOBID PARTITION     NAME     USER ST          START_TIME  NODES SCHEDNODES           NODELIST(REASON)
          17457878       acc LLM_task nct01232 PD                 N/A      1 (null)              (None)
```

```
[nct01232@alogin1 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          17457878       acc LLM_task nct01232  R       0:44      1 as01r2b24
```

4. Model Precision

Resultado de ejecución de **submit.LLM.task4.slurm**

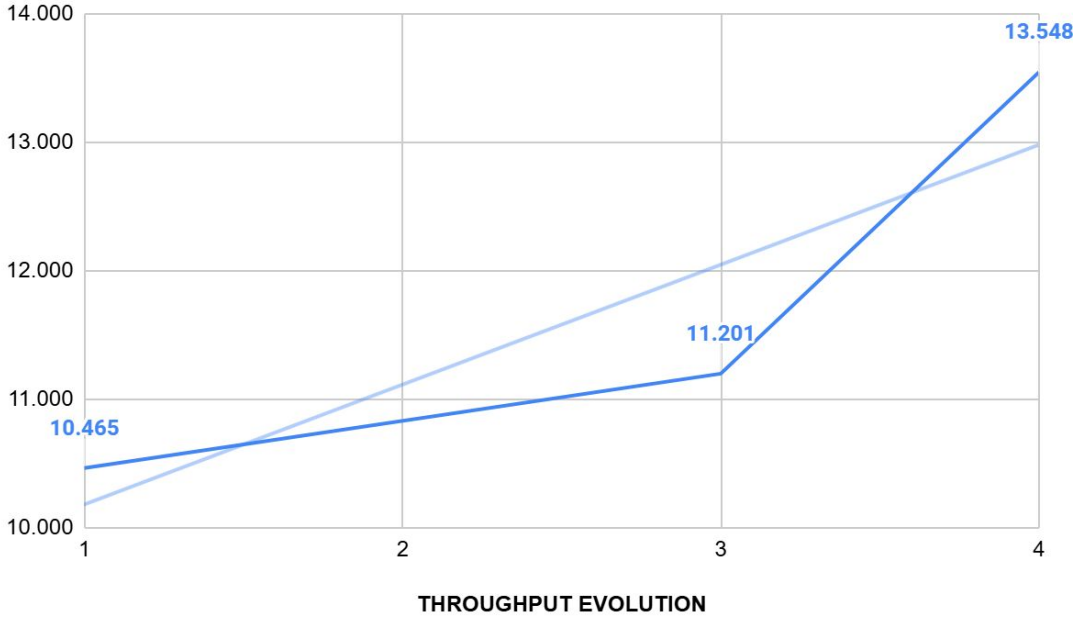
TASK	Job ID	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger	
1	17261309	1	6	10.465	20	fp32	NO	eager	FALSE	Baseline
2	17454556	1	7	N/A	61'50	fp32	NO	eager	FALSE	OOM-Límite
3	17455580	1	5	11.200'76	58'27	fp32	bf16	eager	FALSE	Mixed Precision
4	17457878	1	5	13.547'90	47'66	bf16	bf16	eager	FALSE	Model Precision bf16

- **Throughput mejora un 21%** respecto a Task 3
- **Max Reserved Memory reduce un 18’2%** respecto a Task 3

Deberíamos poder subir BATCH SIZE ya que aún estamos lejos del OOM

4. Model Precision

Progreso del Throughput



¿Y si se cambia **MICRO_BATCH_SIZE** de 5 a 7?

```
##############################################
#### Variable to be modified by the student
##############################################

MICRO_BATCH_SIZE=7
```

5. Increasing Batch Size

```
[nct01232@alogin1 ~]$ sbatch submit.LLM.task5.slurm
Submitted batch job 17462958
[nct01232@alogin1 ~]$ squeue --start
             JOBID PARTITION     NAME     USER ST          START_TIME  NODES SCHEDNODES           NODELIST(REASON)
          17462958       acc LLM_task nct01232 PD                 N/A      1 (null)              (None)
```

```
[nct01232@alogin1 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          17462958       acc LLM_task nct01232  R       4:01      1 as02r2b23
[nct01232@alogin1 ~]$
```

5. Increasing Batch Size

Igual que en task anterior pero **MICRO_BATCH_SIZE** pasa de 5 a 7

TASK	Job ID	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger	
1	17261309	1	6	10.465	20	fp32	NO	eager	FALSE	Baseline
2	17454556	1	7	N/A	61'50	fp32	NO	eager	FALSE	OOM-Límite
3	17455580	1	5	11.200'76	58'27	fp32	bf16	eager	FALSE	Mixed Precision
4	17457878	1	5	13.547'90	47'66	bf16	bf16	eager	FALSE	Model Precision bf16
5	17461753	1	7	14.291'11	61'35	bf16	bf16	eager	FALSE	Aumento BS

- 5’5% Mejora del throughput respecto a MBS = 5
- Uso de memoria muy cercano al límite. MBS = 7 es el máximo

6. Enabling Flash Attention

Liger Kernel (LIGER_KERNEL=true) para reducir el uso de memoria y aumentar el Training Throughput.

sbatch submit.LLM.task6.slurm

```
[nct01232@alogin1 results]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          17464732       acc LLM_task nct01232  R       2:22      1 as03r1b17
```

6. Enabling Flash Attention

Resultados

TASK	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger	
1	1	6	10.465	20	fp32	NO	eager	FALSE	Baseline
2	1	7	N/A	61'50	fp32	NO	eager	FALSE	OOM-Límite
3	1	5	11.200'76	58'27	fp32	bf16	eager	FALSE	Mixed Precision
4	1	5	13.547'90	47'66	bf16	bf16	eager	FALSE	Model Precision bf16
5	1	7	14.291'11	61'35	bf16	bf16	eager	FALSE	Aumento BS
6	1	7	24.396'41	39'67	bf16	bf16	Flash Attention	FALSE	Flash Attention

- Casi el doble de Throughput
- Casi la mitad de Memoria reservada

7. Increasing Batch Size with Flash Attention

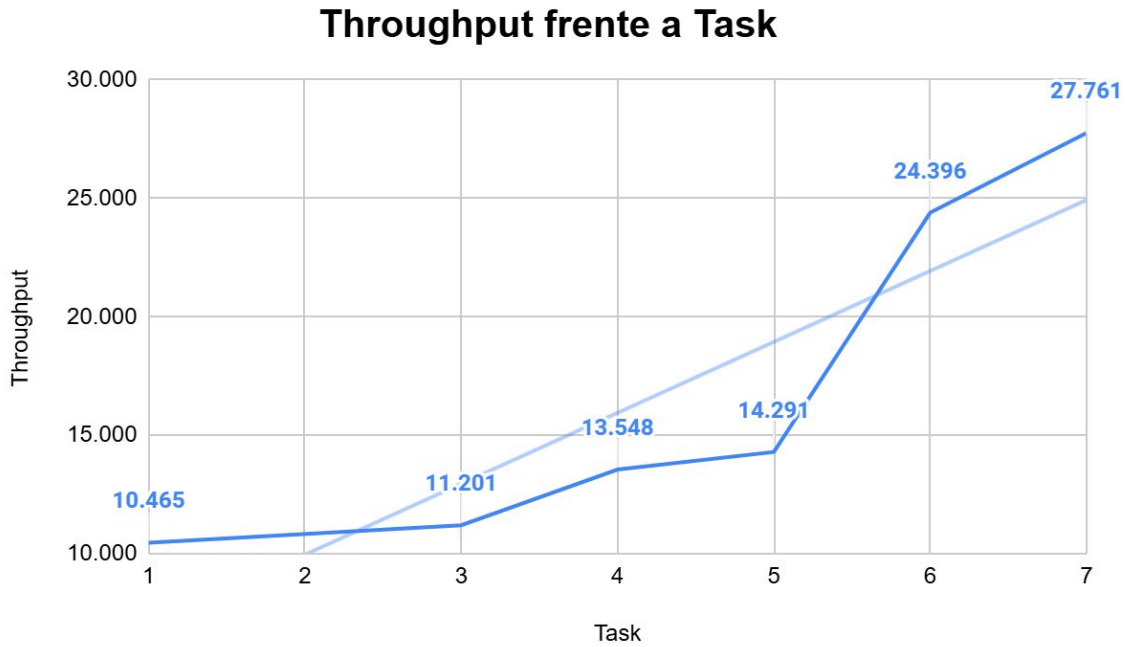
MICRO_BATCH_SIZE = 14

TASK	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger	
1	1	6	10.465	20	fp32	NO	eager	FALSE	Baseline
2	1	7	N/A	61'50	fp32	NO	eager	FALSE	OOM-Límite
3	1	5	11.200'76	58'27	fp32	bf16	eager	FALSE	Mixed Precision
4	1	5	13.547'90	47'66	bf16	bf16	eager	FALSE	Model Precision bf16
5	1	7	14.291'11	61'35	bf16	bf16	eager	FALSE	Aumento BS
6	1	7	24.396'41	39'67	bf16	bf16	sdpa	FALSE	Flash Attention
7	1	14	27.760'68	60'99	bf16	bf16	sdpa	FALSE	Larger BS

7. Increasing Batch Size with Flash Attention

- **13’4% aumento throughput** respecto a task 6
- Uso de **memoria cercano al límite**
- MICRO_BATCH_SIZE = 14 es el **máximo** para la configuración actual

7. Increasing Batch Size with Flash Attention



8. Using the Liger kernel

Se habilita Liger Kernel estableciendo **LIGER_KERNEL = true** en el script de slurm.

Por lo demás, el modelo es el mismo que el anterior.

sbatch submit.LLM.task8.slurm

```
-rw-r--r-- 1 nct01232 nct 39602 Mar 21 19:57 R-LLM_task8.17470773.err
-rw-r--r-- 1 nct01232 nct  2802 Mar 21 19:57 R-LLM_task8.17470773.out
```

8. Using the Liger kernel

submit.LLM.task8.slurm

TASK	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger	
1	1	6	10.465	20	fp32	NO	eager	FALSE	Baseline
2	1	7	N/A	61'50	fp32	NO	eager	FALSE	OOM-Límite
3	1	5	11.200'76	58'27	fp32	**bf16**	eager	FALSE	Mixed Precision
4	1	5	13.547'90	47'66	**bf16**	**bf16**	eager	FALSE	Model Precision bf16
5	1	7	14.291'11	61'35	**bf16**	**bf16**	eager	FALSE	Aumento BS
6	1	7	24.396'41	39'67	**bf16**	**bf16**	**sdpa**	FALSE	Flash Attention
7	1	14	27.760'68	60'99	**bf16**	**bf16**	**sdpa**	FALSE	Larger BS
8	1	14	36.777'21	32'45	**bf16**	**bf16**	**sdpa**	**TRUE**	Liger Kernel activated

- 32’92% de aumento Throughput
- 46’81% de reducción de uso de memoria

8. Using the Liger kernel

submit.LLM.task8.slurm

TASK	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger	
1	1	6	10.465	20	fp32	NO	eager	FALSE	Baseline
2	1	7	N/A	61'50	fp32	NO	eager	FALSE	OOM-Límite
3	1	5	11.200'76	58'27	fp32	**bf16**	eager	FALSE	Mixed Precision
4	1	5	13.547'90	47'66	**bf16**	**bf16**	eager	FALSE	Model Precision bf16
5	1	7	14.291'11	61'35	**bf16**	**bf16**	eager	FALSE	Aumento BS
6	1	7	24.396'41	39'67	**bf16**	**bf16**	**sdpa**	FALSE	Flash Attention
7	1	14	27.760'68	60'99	**bf16**	**bf16**	**sdpa**	FALSE	Larger BS
8	1	14	36.777'21	32'45	**bf16**	**bf16**	**sdpa**	**TRUE**	Liger Kernel activated

- 32’92% de aumento Throughput
- 46’81% de reducción de uso de memoria

Mucho margen en cuanto a memoria!

9. Augmenting batch size due to Liger kernels

Como hay mucho margen con la memoria, puede aumentarse el MBS

```
[nct01232@alogin1 ~]$ sbatch submit.LLM.task9.slurm
Submitted batch job 17477502
[nct01232@alogin1 ~]$ squeue --start
             JOBID PARTITION     NAME     USER ST          START_TIME  NODES SCHEDNODES           NODELIST(REASON)
          17477502       acc LLM_task nct01232 PD                 N/A      1 (null)              (Resources)
[nct01232@alogin1 ~]$
```

```
[nct01232@alogin1 results]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          17477502       acc LLM_task nct01232  R       6:02      1 as03r1b12
```

9. Augmenting batch size due to Liger kernels

MICRO_BATCH_SIZE = 37 (264% más)

TASK	GPUs	Batch Size	Tr.Throughput (tk/s/GPU)	Memory (GiB)	Model Precission	Mixed Precission	Attention type	Liger	
1	1	6	10.465	20	fp32	NO	eager	FALSE	Baseline
2	1	7	N/A	61'50	fp32	NO	eager	FALSE	OOM-Límite
3	1	5	11.200'76	58'27	fp32	**bf16**	eager	FALSE	Mixed Precision
4	1	5	13.547'90	47'66	**bf16**	**bf16**	eager	FALSE	Model Precision bf16
5	1	7	14.291'11	61'35	**bf16**	**bf16**	eager	FALSE	Aumento BS
6	1	7	24.396'41	39'67	**bf16**	**bf16**	**sdpa**	FALSE	Flash Attention
7	1	14	27.760'68	60'99	**bf16**	**bf16**	**sdpa**	FALSE	Larger BS
8	1	14	36.777'21	32'45	**bf16**	**bf16**	**sdpa**	**TRUE**	Liger Kernel activated
9	1	37	48.098'99	62'34	**bf16**	**bf16**	**sdpa**	**TRUE**	Larger BS with Kernel

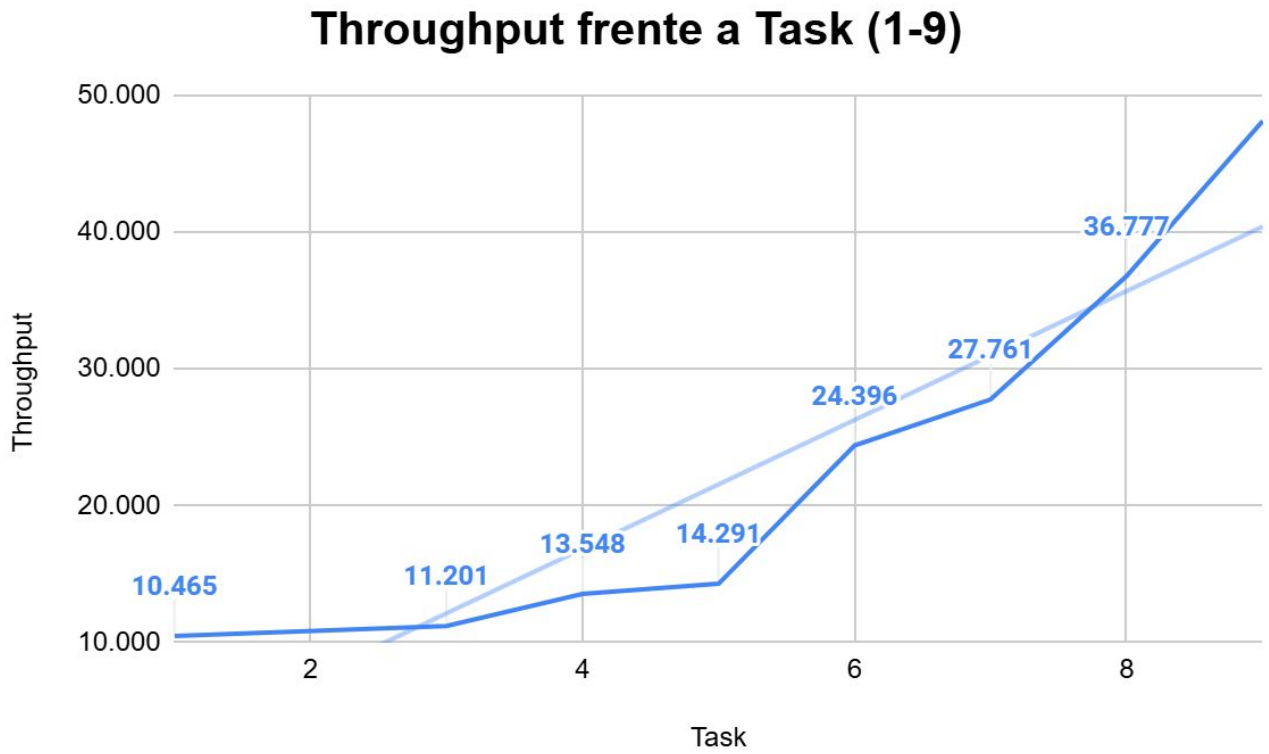
9. Augmenting batch size due to Liger kernels

MICRO_BATCH_SIZE = 37 (264% más)

- **Throughput aumentó en un 30’8%** respecto a task 8
- **Uso de memoria aumentó un 92’14%** respecto a task 8
- Max Reserved GPU **Memory = 62’34 GiB**

Muy cerca del límite en cuanto a memoria

9. Tasks 1-9



459% más Th.

616% más mem.

Sin OOM

¿Y si escalamos esto usando más GPUs?

10. Scaling on multiple GPUs

Aumentar cantidad de GPUs

Se usarán **1, 2, 4, 8, 16 y 32 GPUs**

```
[nct01232@alogin1 ~]$ ls -l submit.LLM.task10*
-rw-r--r-- 1 nct01232 nct 3270 Mar 19 17:50 submit.LLM.task10_16GPU.slurm
-rw-r--r-- 1 nct01232 nct 3309 Mar 19 17:50 submit.LLM.task10_1GPU.slurm
-rw-r--r-- 1 nct01232 nct 3272 Mar 19 17:50 submit.LLM.task10_2GPU.slurm
-rw-r--r-- 1 nct01232 nct 3270 Mar 19 17:51 submit.LLM.task10_32GPU.slurm
-rw-r--r-- 1 nct01232 nct 3272 Mar 19 17:50 submit.LLM.task10_4GPU.slurm
-rw-r--r-- 1 nct01232 nct 3272 Mar 19 17:50 submit.LLM.task10_8GPU.slurm
```

```
[nct01232@alogin1 ~]$ sbatch submit.LLM.task10_32GPU.slurm
Submitted batch job 17498018
```

10. Scaling on multiple GPUs

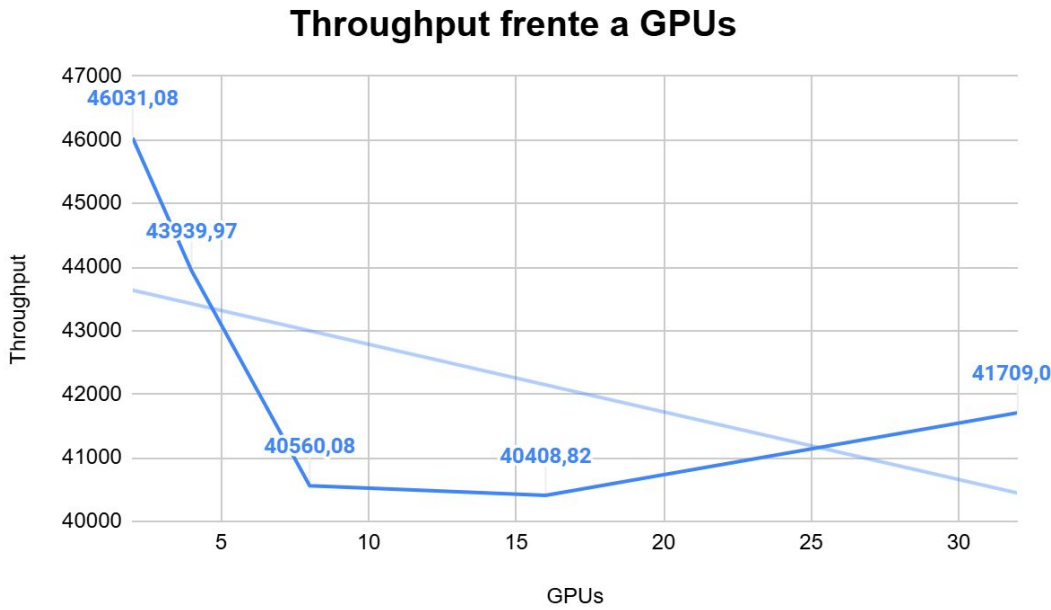
Tras terminar las ejecuciones con distinta cantidad de GPUs:

```
[nct01232@alogin1 results]$ ls R-LLM_task10_*
R-LLM_task10_16GPU.17497180.err  R-LLM_task10_2GPU.17491926.err   R-LLM_task10_4GPU.17494492.err
R-LLM_task10_16GPU.17497180.out  R-LLM_task10_2GPU.17491926.out   R-LLM_task10_4GPU.17494492.out
R-LLM_task10_1GPU.17488445.err   R-LLM_task10_32GPU.17498018.err  R-LLM_task10_8GPU.17496359.err
R-LLM_task10_1GPU.17488445.out   R-LLM_task10_32GPU.17498018.out  R-LLM_task10_8GPU.17496359.out
[nct01232@alogin1 results]$
```

10. Scaling on multiple GPUs

Throughput con distinta cantidad de GPUs

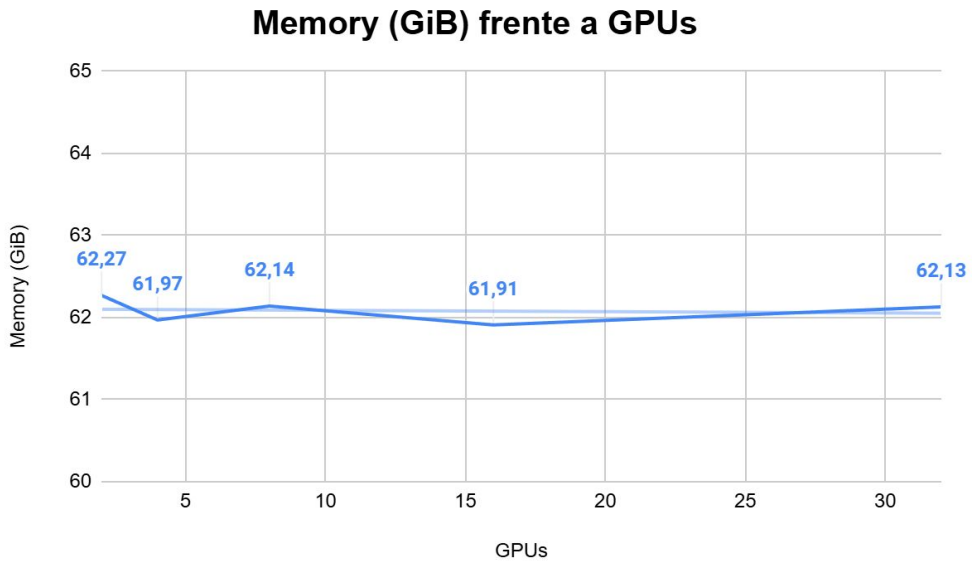
GPUs	Throughput	Memory (GiB)
2	46031,08	62,27
4	43939,97	61,97
8	40560,08	62,14
16	40408,82	61,91
32	41709,05	62,13



10. Scaling on multiple GPUs

Memoria con distinta cantidad de GPUs

GPUs	Throughput	Memory (GiB)
2	46031,08	62,27
4	43939,97	61,97
8	40560,08	62,14
16	40408,82	61,91
32	41709,05	62,13

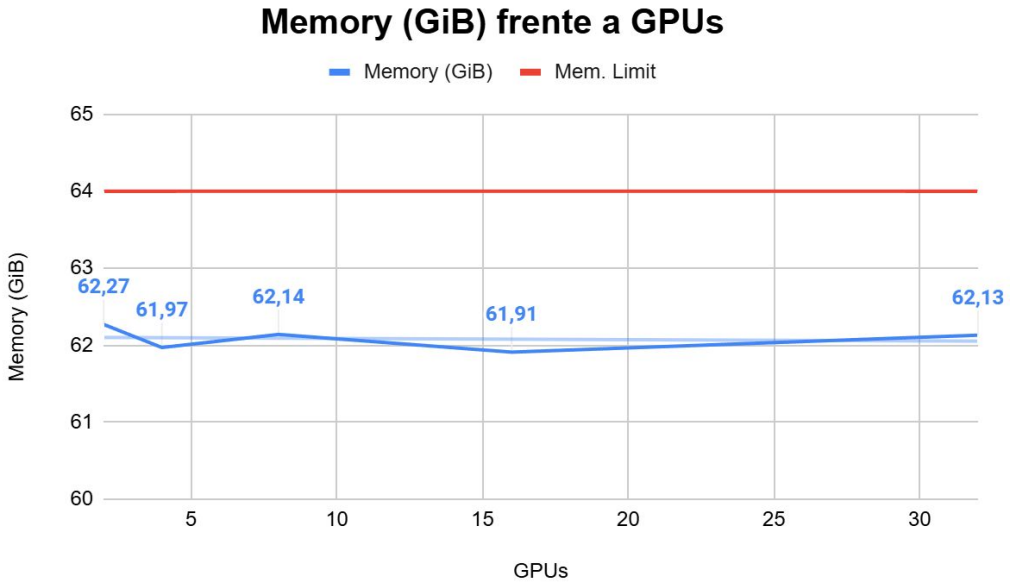


10. Scaling on multiple GPUs

Explotando al máximo la memoria disponible

El límite lo establece la H100

GPUs	Throughput	Memory (GiB)
2	46031,08	62,27
4	43939,97	61,97
8	40560,08	62,14
16	40408,82	61,91
32	41709,05	62,13



11. Conclusions

1. Mixed Precision, Model Precision bf16, Flash Attention y Liger Kernel han permitido **mejorar el Th. en 459%.**
2. Las **optimizaciones también permiten reducir el uso de memoria** y aumentar el MICRO_BATCH_SIZE.
3. El **escalado** de GPUs **no aumenta el Th.** De hecho, tiende a disminuir.
4. El **escalado** de GPUs **mantiene el uso de memoria** prácticamente constante, muy cercano al límite.

Gracias