Sobirania personal, IA local

O com evitar que ens robin les idees

Martí SC — 2026-09-26

Per què IA local?

- **Privadesa / sobirania**: les dades no surten de la màquina
- **Cost**: sense pagar per cada token generat
- **Sense connexió**: no depèn d'un proveïdor extern
- **Control**: ajust fi i possibilitat de canviar de model

> **Ara bé**: els models capdavanters encara guanyen en tasques de raonament molt complexes — *per ara*.

Les preguntes

- Què podem executar?
- Quina serà la velocitat de resposta?
- Què estem perdent?

Quina VRAM teniu?

Memòria	Exemples de GPU
8 GB	5050, 5060, 4060
12 GB	5070, 4070, 2060*, 3060*, RX 6700
16 GB	4060 Ti, 4080, 5080, RX 6800
24 GB	3090, 4090, RX 7900 XTX
32 GB	5090
Més de 32 GB	Mac / DGX Spark

Models de pesos oberts

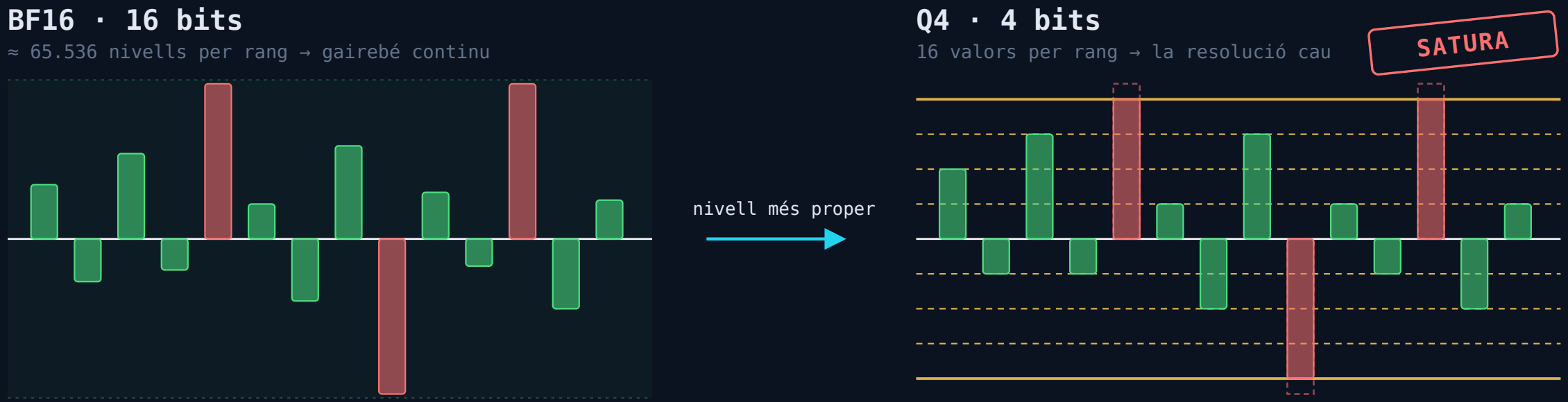
Model	Mides	Orientació
Qwen3.X (Alibaba)	8B–235B	Tot terreny
GLM 5.3 (Zhipu)	321B–753B	Frontera
Kimi K3 (Moonshot)	2,8T MoE (105B actius)	El més gran de pesos oberts
DeepSeek V4	290B–1,6T	Massa gran per a nosaltres

7–14B: bon assistent · 27–70B: agent per a tasques complexes

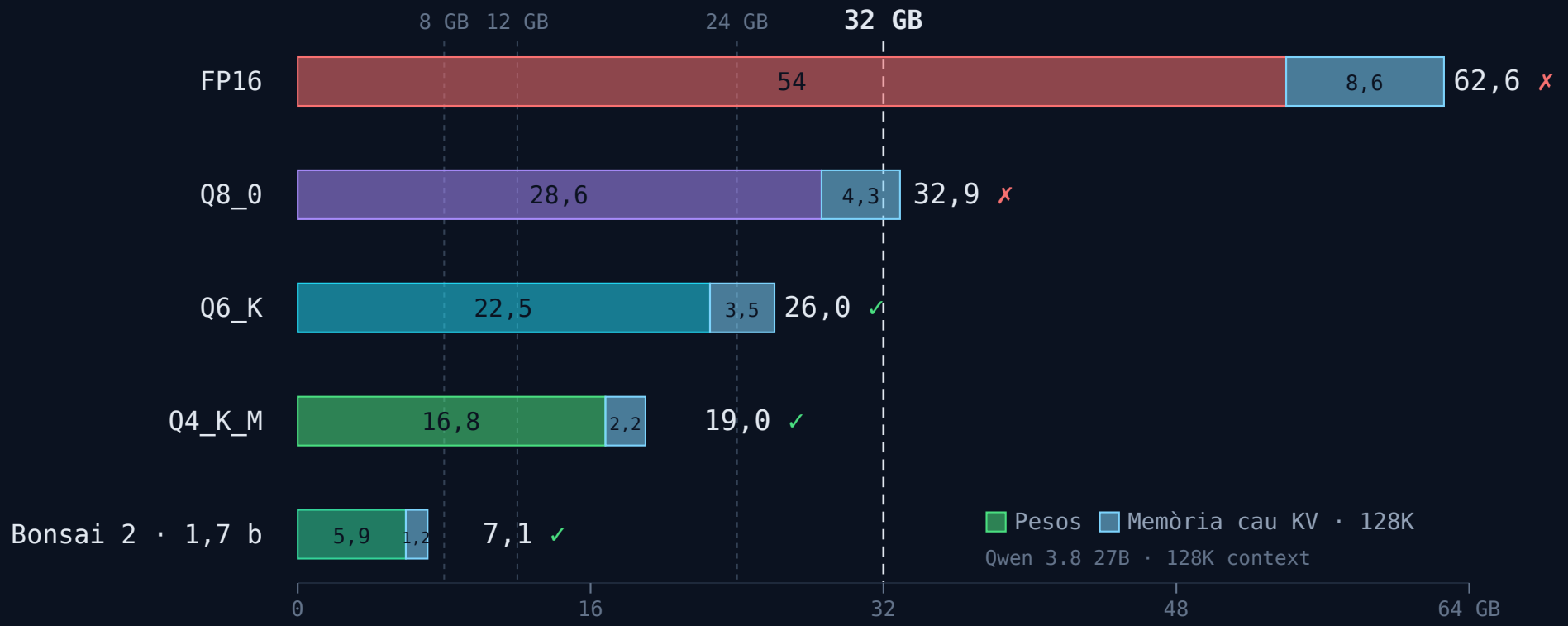
Quantització: comprimir el model

BF16 és la precisió original: `27B × 16 bits ≈ 54 GB`

BF16 → INT8, INT4… o ternari `{-1, 0, +1}`



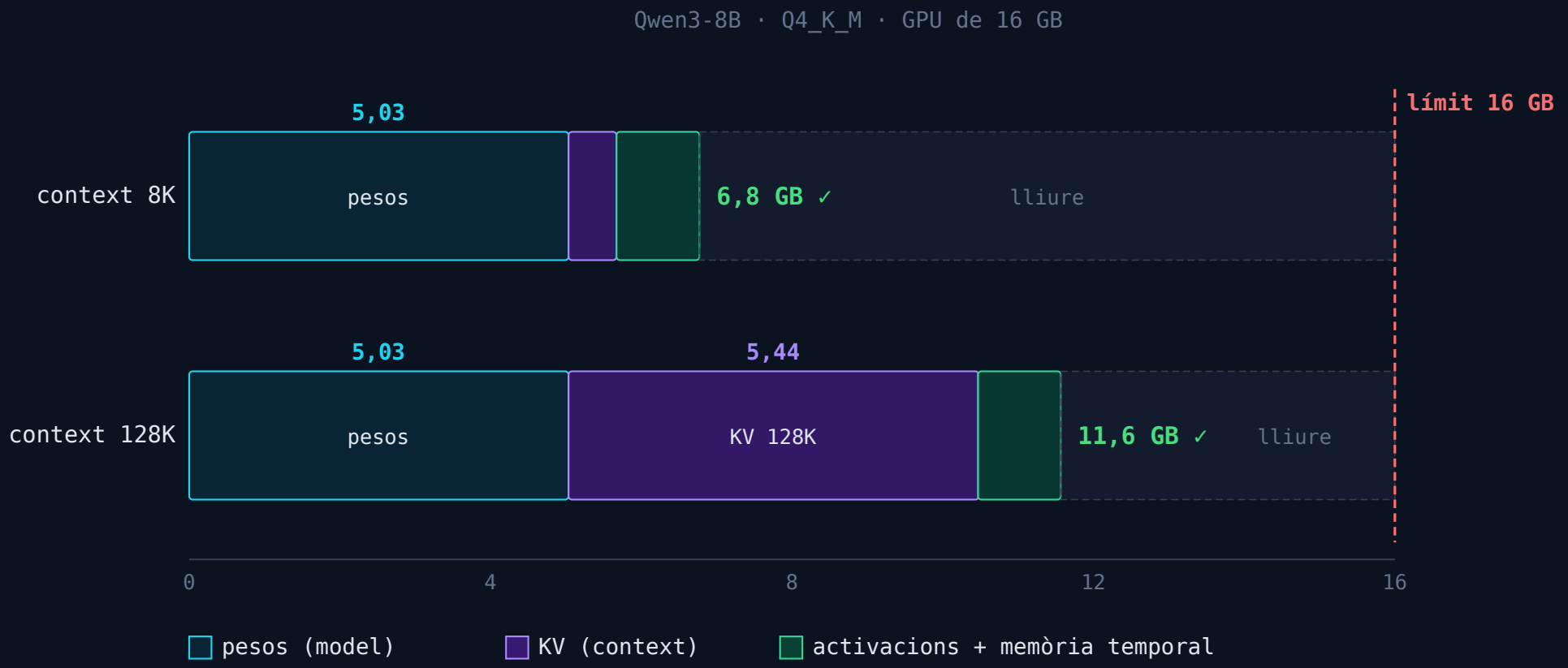
Hem d'usar models quantitzats



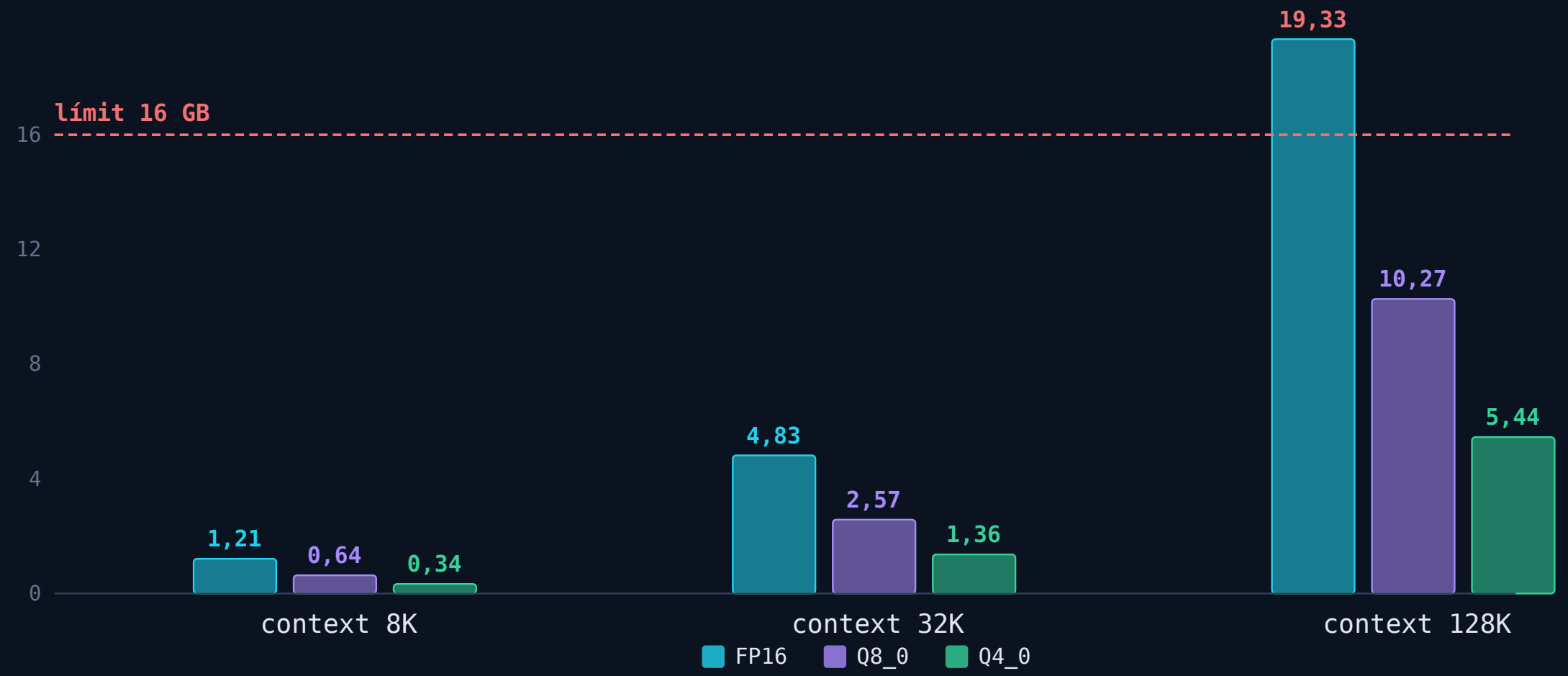
GGUF: Q8, Q6, Q4… · **NVFP4**: per a les noves NVIDIA · **Ternari**: menys de 2 bits

Conseqüències

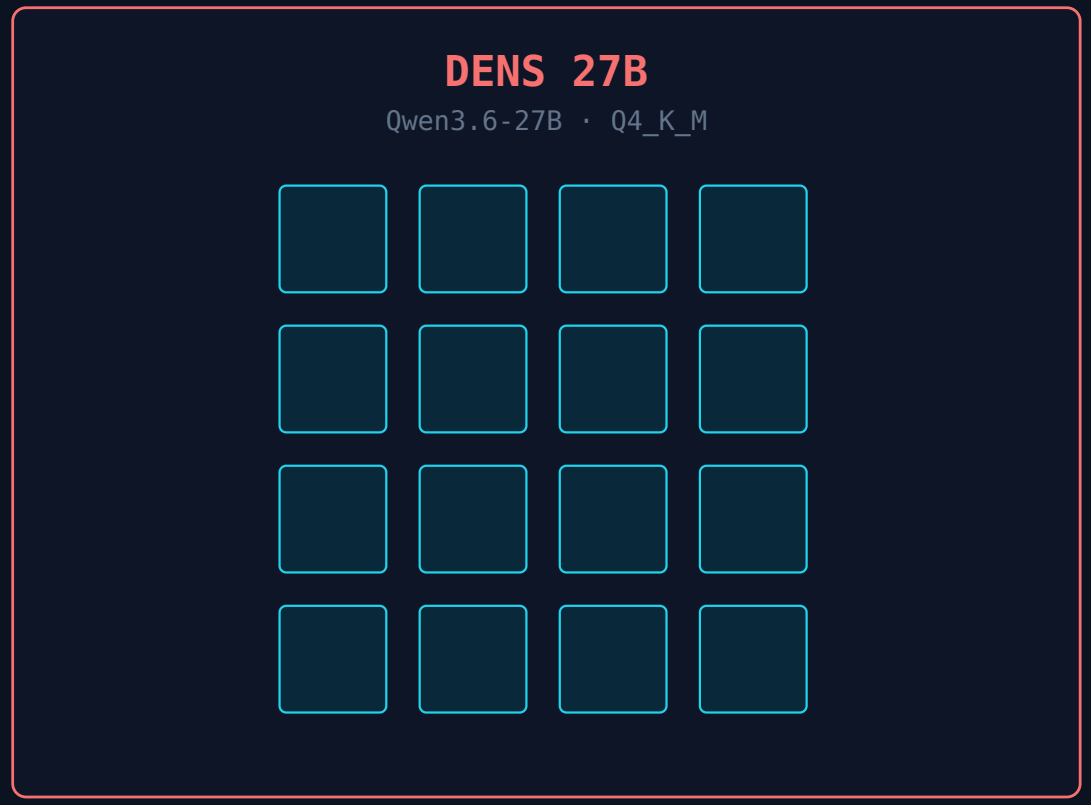
On va la memòria?



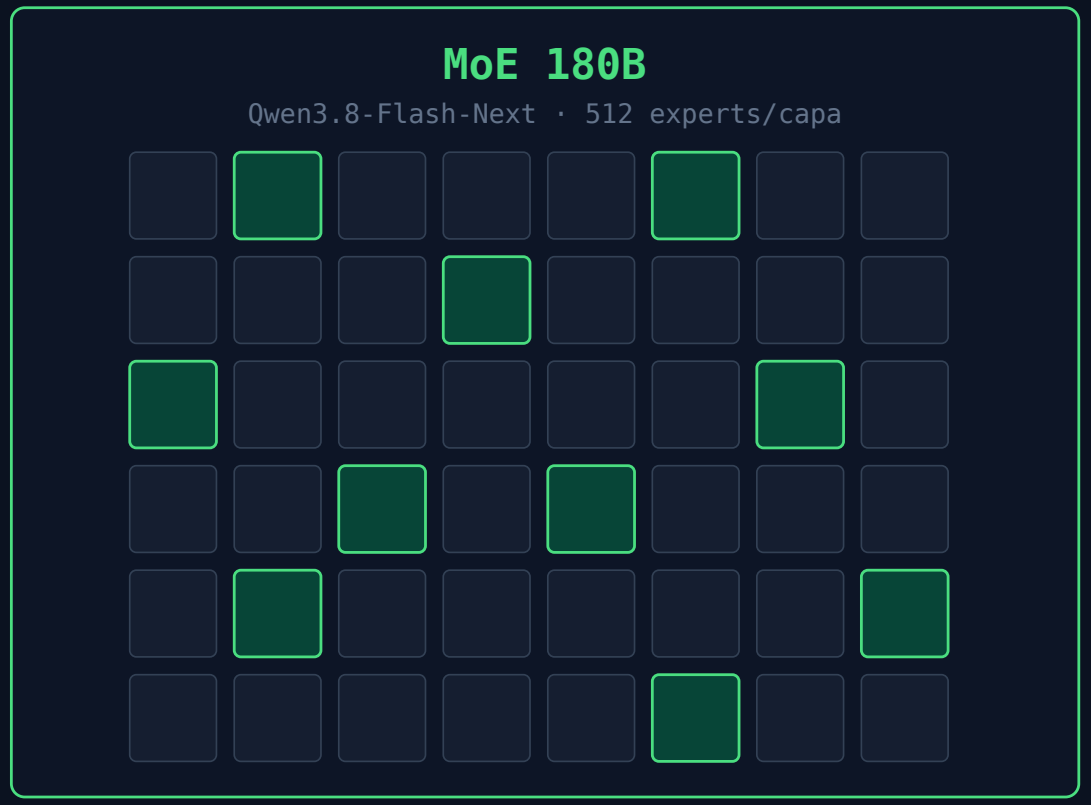
La memòria cau KV limita el context



Model dens o MoE: què processa cada token?



Tots els 16 blocs actius per token (100%)

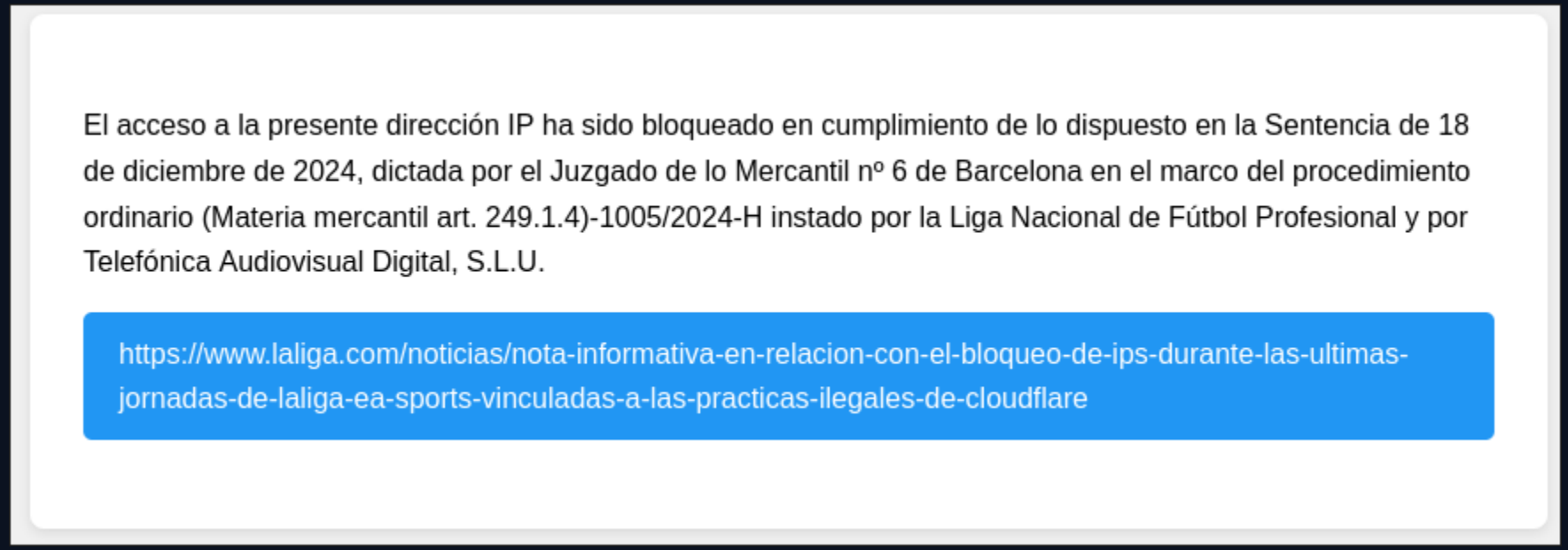


Per token: 10 + 1 expert actius ≈ 6B p = 3% del total

Un model gran pot combinar VRAM i RAM; l'SSD serveix per desar-lo i carregar-lo.

Censura de LaLiga

En l'explicació de Qwen3.8-Flash-Next apareix això, de cop:



La GPU fa brum brum!

Per una resposta de 200 paraules:

Velocitat de generació	Temps	Sensació
5 tok/s	53 s	Extremadament lent
10 tok/s	27 s	Lent
20 tok/s	13 s	Correcte, però pot frustrar
40 tok/s	7 s	Acceptable
80 tok/s	3 s	Gairebé instantani

Processar el prompt ≠ generar la resposta

```
Inferència = prefill (processar l'entrada) + decoding (generar la sortida)
```

	Prefill	Decoding
Què passa?	Processa els tokens d'entrada	Genera un token per pas i el torna a introduir al model
Limitació habitual	**Càlcul**	**Amplada de banda de la memòria**
Mètrica útil	TTFT: temps fins al primer token	**tok/s**
Percepció	Temps fins que comença la resposta	Fluïdesa de la resposta

- La velocitat de generació pot baixar quan creix el context.

Com accelerem la generació?

Tècnica	Com funciona?	Què millora?
Generació especulativa	Un model petit proposa diversos tokens i el model principal els verifica en paral·lel	Latència i tok/s d'una conversa
MTP (predicció de diversos tokens)	El mateix model té mòduls entrenats per predir tokens futurs	Pot accelerar la conversa sense un altre model
Espais de conversa	Diverses converses simultànies, limitades per la memòria cau KV	Concurrència
Agrupació contínua	Agrupa peticions de diverses converses	Rendiment total

> ⚠️ Proposar i verificar tokens té un cost; el guany depèn de quants se n'accepten.

Una conversa: generació especulativa o MTP · *Moltes converses*: agrupació contínua.

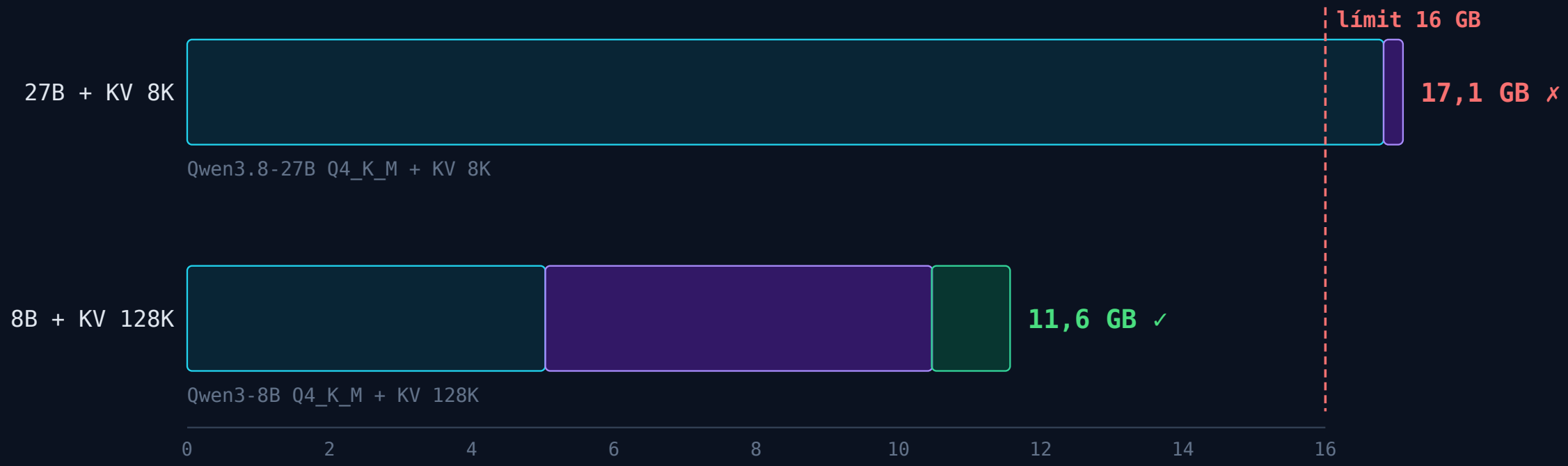
El cost del context llarg

Durant la generació, cada token consulta la memòria cau KV anterior.

Context	Velocitat aproximada
8K	referència
32K	-6%
128K	-20%
256K	**-34%**

Un context llarg **pot reduir la velocitat**, i pocs ho analitzen.

Què hi cap a la vostra GPU?



- Fem números: pesos + memòria cau KV + 1–3 GB de marge
- Després cal mesurar-ne la velocitat

Model × quantització × GPU

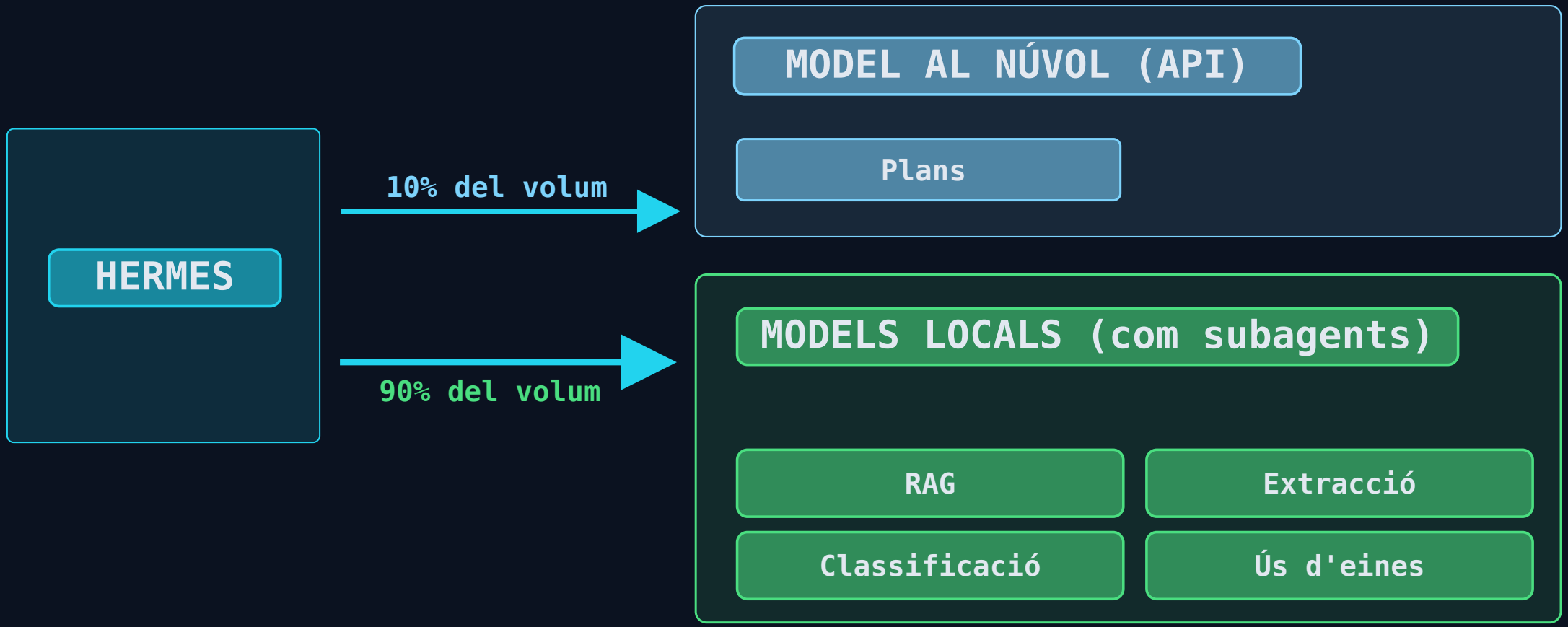
GPU / RAM	Exemples de models	Context orientatiu	Notes
12 GB	8B Q4	8–32K	27B a 2 bits: 16K amb marge estret
16 GB	8B Q4 · 14B Q4	8B: 128K* · 14B: 64K*	*YaRN + KV Q4_0
24 GB	14B Q4 · 27B Q4_K_M	Fins a 128K amb el 27B	KV Q8_0
32 GB	32B Q4 · 27B UD-Q4_K_XL	Fins a 256K amb el 27B	KV Q8_0
5090 + RAM	32B o **MoE 111 GB**	48K+	`-ncpu-moe`
Mac 128 GB	70B, o MoE	llarg	unificat, més lent

Com ho executem?

Eina	Què aporta?	Tria-la si...
LM Studio	Interfície gràfica i cercador de models	Prefereixes una interfície gràfica
Ollama	Simplifica l'ús de llama.cpp	Vols començar ràpid
llama.cpp	GGUF, CPU/GPU, MTP, VRAM/RAM	Control i flexibilitat
MLX	Optimitzat per a Apple Silicon	Tens un Mac
vLLM	Servei eficient i moltes peticions simultànies	Tens diversos usuaris / subagents

Poden oferir una API compatible amb la d'OpenAI.

Arquitectura híbrida



Conclusions

1. **La IA local val la pena provar-la.**
   Jugueu-hi tenint presents les seves limitacions.
2. **La VRAM determina què hi cap.**
   L'amplada de banda condiciona la velocitat de resposta.
3. **No hi ha cap benchmark universal.**
   Mesureu el vostre cas real: no us refieu de l'expectació ni d'una xifra aleatòria de les xarxes. I torneu-ho a provar aviat: tot evoluciona molt de pressa.

Preguntes?