

Implementació i avaluació de rendiment d'un sistema d'IA basat en models LLMs en un entorn HPC

Ángel Blanco Pereira

Resum— Davant el perill d'utilitzar una intel·ligència artificial en línia a causa d'una possible pèrdua de dades o el tractament de dades privades, el Centre Nacional d'Anàlisi Genòmic (CNAG) ha decidit optar per usar el seu clúster de GPU i CPU per tenir una IA local per satisfer les seves necessitats dintre dels seus projectes. Per poder dur a terme aquesta feina, he fet una investigació sobre com fer servir de la millor manera els recursos disponibles, millors softwares LLM en l'àmbit de rendiment, eficiència, maneres d'implementació,... Finalment, es fan diferents proves de rendiment del model creat on s'avalua factors com quantitat d'accions humanes requerides, correctesa de les solucions o respostes, velocitat, entre altres i serà comparat amb models en línia coneguts obtenint finalment la conclusió de si és rendible i viable la creació d'aquest model per a la companyia.

Paraules clau— IA, LLM, CNAG, CPU, GPU, NPU, Cluster, Workstation, Hardware, Software, data breach, Apache 2.0, MIT, On-Premise.

Abstract— Given the risks of using online Artificial Intelligence—specifically regarding potential data loss or the exposure of sensitive information the National Center for Genomic Analysis (CNAG) has opted to leverage its own GPU and CPU cluster to implement a local AI solution tailored to its internal project requirements. To achieve this, I have conducted research on optimizing the available resources and identifying the best LLM frameworks in terms of performance, efficiency, and deployment strategies. Finally, various performance benchmarks were conducted to evaluate the local model, focusing on factors such as human intervention requirements, accuracy of solutions, and inference speed. These results were compared against well-known commercial online models to determine the viability and cost-effectiveness of maintaining an in-house AI infrastructure for the organization.

Index Terms— AI, LLM, CNAG, CPU, GPU, NPU, Cluster, Workstation, Hardware, Software, data breach, Apache 2.0, MIT, On-Premise.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

AQUEST document és un resum del treball d'investigació efectuat per ajudar al Centre Nacional d'Anàlisi Genòmica (CNAG) [3].

L'auge de la intel·ligència artificial generativa ha transformat la manera en què les organitzacions gestionen el coneixement i el desenvolupament de programari. No obstant això, els centres de recerca d'alt nivell com CNAG o altres organitzacions amb dades molt crítiques com asseguradores o organitzacions governamentals no els hi resulta tan fàcil l'adopció d'aquestes eines comercials basades al núvol com ara ChatGPT o Claude, ja que presenta barreres crítiques en relació amb la seguretat i la sobirania de les dades.

1.1 El Centre Nacional d'Anàlisi Genòmic (CNAG)

El CNAG és una institució de referència internacional dedicada a l'anàlisi genòmica a gran escala. Dit d'una altra manera, les seves activitats estan relacionades amb el tractament, gestió, entre altres de l'ADN i dades clíniques altament sensibles i protegides per normatives de

privacitat. Per tant, qualsevol filtració o data breach [2] d'aquestes dades o la seva obtenció pot comportar conseqüències legals i ètiques molt greus.

1.2 La Problemàtica: IA i Privacitat

Actualment, dintre d'aquest tipus d'indústries amb dades crítiques es troben davant un dilema, l'ús de models d'IA en línia poden accelerar molt la feina, però com a conseqüència, s'ha d'assumir el risc de transferir dades privades a servidors externs. Llavors, la manca d'una alternativa local segura pot augmentar la distància entre competidors en velocitat o desenvolupament entre centres que han integrat aquesta eina i els que no.

1.3 Requisits de la solució

A causa de l'alt nivell requerit per a un centre com el CNAG, aquest treball investiga la implementació d'una infraestructura d'IA local operativa dintre del clúster [1] local del centre, complint els estàndard establerts pel centre en diferents seccions:

- Privacitat Absoluta, les dades no surten mai de la

xarxa.

- Rendiment i Eficiència, utilitzar el sistema més adient al clúster.
- Cost i qualitat/correctesa, trobar un equilibri entre tenir un cost baix i que les respostes obtingudes siguin correctes i el més òptim possible.

1.4 Organització del document

El document se separa en les següents seccions: Objectius, metodologia i planificació, estat de l'art, investigació inicial, proves i conclusió.

Finalment, hi ha una secció per l'ús d'IA, agraïments i bibliografia que recopila informació varia.

2 OBJECTIUS

L'objectiu d'aquest treball no es limita a la instal·lació d'un programari, sinó a l'avaluació crítica d'una infraestructura d'autoconsum d'IA en un entorn d'alta computació.

2.1 Objectiu general

L'objectiu principal d'aquesta investigació és saber si és viable o possible utilitzar el clúster disponible dintre del centre com a host d'un Model de Llenguatge Gran (LLM [4]) amb el propòsit d'establir un entorn d'execució local que garanteixi també la seguretat de les dades, eliminant els riscos de data breach i filtracions de propietat intel·lectuals o dades privades.

2.2 Objectius específics

Per poder assolir l'objectiu inicial, s'han definit les següents fites tècniques:

- **Anàlisi i adequació de l'arquitectura de hardware:** Fer una investigació inicial on podem arribar a resoldre les diferents preguntes plantejades per identificar quins punts claus ajuden a obtenir millores en el rendiment.
- **Selecció i optimització del stack de software:** Investigar i seleccionar el framework d'inferència (Open LLM) que ofereixi la millor ràtio de tokens per segon tenint en compte l'arquitectura de memòria i processament del CNAG.
- **Benchmarking comparatiu de rendiment:** Fer proves de càrrega i velocitat per comparar latència i la precisió de les respostes del LLM local enfront d'alternatives en línia.
- **Avaluació de la qualitat i seguretat de les solucions:** Mesurar la correctesa de les respostes del model local i la capacitat per a mantenir la integritat de les dades crítiques del centre durant la utilització de l'eina.
- **Escollir de manera definitiva els components que formaran el producte definitiu:** Treure les conclusions de les dades obtingudes als benchmarkings per construir el model definitiu amb la

possibilitat de la implementació final del model.

- **Generació d'un informe detallat del treball:** Redactar un informe corporatiu final que defineixi el procediment a l'hora d'implementar el sistema.

3 METODOLOGIA

3.1 Enfocament de la investigació

Aquesta investigació adopta un enfocament quantitatiu i corporatiu, ja que, l'objectiu és caracteritzar de manera empírica el rendiment del clúster situat al CNAG en diferents àmbits numèrics com temps, consum,... a l'hora d'executar software d'IA i comparar-ho amb models populars.

3.2 Disseny d'investigació (procediment experimental)

L'estudi realitzat es divideix en quatre etapes seqüencials encarregades de definir la proposta en l'àmbit tècnic.

- **Auditoria del hardware i software:** Catalogar els recursos disponibles dintre del clúster pel que fa a CPUs, GPUs i RAM per saber les restriccions de memòria que seran el principal limitant del model que s'utilitzarà.
- **Configuració de l'entorn de desplegament:** Implementació dels entorns virtuals per poder instal·lar el framework de inferència com Ollama, vLLM o llama.cpp, buscant compatibilitat entre les arquitectures del centre.
- **Executar benchmarks per avaluar el rendiment:** Fer test amb sets de preguntes estandaritzades per avaluar el model d'acord amb diferents punts definits més endavant.
- **Anàlisi de les dades i validació:** Processar els resultats amb taules comparatives i gràfics de rendiment per avaluar el model a l'hora de complir els requisits del centre d'acord amb la qualitat i seguretat requerida.

3.3 Col·laboracions

Dintre de la investigació s'ha col·laborat amb el CNAG on a més de proporcionar-me l'estructura, també han ofert el suport tècnic necessari per alinear la investigació amb les necessitats reals de l'entitat, tot a través de la creació d'un compte on puc accedir al clúster per fer les proves necessàries. Com a complement, s'han creat també canals de comunicació aïllats dels utilitzats dintre del centre on compartim les diferents investigacions i avenços.

3.4 Definició i volum de les proves

Per tal que l'avaluació sigui estadísticament significativa, no es duran a terme proves aïllades o d'execució única, sinó un conjunt de benchmarks sistemàtics. Es realitzaran un total de 40 proves tècniques per combinació d'elements, 10 per cada bloc de proves que considera el CNAG i jo com a útils per comparar el model local amb les alternatives cloud (com GPT-4o o Claude).

Els blocs de proves són:

- **Extracció de dades estructurades:** Donada una cadena de text en format brut (logs del sistema o dades genòmiques), el model ha d'extreure camps concrets i formatar-los en un JSON vàlid. Es mesurarà la taxa d'error de sintaxi en el JSON.
- **Generació de tests unitaris:** Es proporcionaran funcions complexes de Python/R i el model haurà de generar els tests unitaris corresponents. Es valorarà la cobertura del codi generat.
- **Consultes complexes de bases de dades:** Redacció de consultes SQL o NoSQL a partir de descripcions en llenguatge natural sobre esquemes de dades biomèdiques.
- **Optimització de codi:** Es lliurarà un codi funcional, però ineficient (ex. bucles redundants) perquè el model el refactoritzi buscant millorar la complexitat algorítmica.

Per a cada prova, es recolliran les següents mètriques:

- **Velocitat:** Mesurada en tokens per segon.
- **Temps:** Mesurat en segons.
- **Correctesa:** Validació mitjançant execució del codi.
- **Intervenció humana:** Nombre de correccions (prompts addicionals) necessàries per arribar al resultat correcte.

3.5 Interpretació de les dades obtingudes de les proves

Una vegada fetes les proves, les dades numèriques mesurades seran visibles a taules i/o gràfiques per tal de fer-ho més visible, d'aquesta manera, podrem fer les comparacions entre diferents models de LLM i també amb els models situats al núvol.

4 PLANIFICACIÓ

4.1 Fase d'Anàlisi i Disseny (Mes 1)

- Respondre a les preguntes generades per implementar un model d'IA.
- Definició dels casos d'ús específics per als desenvolupadors dintre del centre.
- Estudi de viabilitat de llicències i privacitat.

4.2 Fase de Desplegament i Benchmarking (Mes 2)

- Configuració de l'entorn i optimització del model.
- Execució de les proves de rendiment.
- Comparativa entre models cloud i local.

4.3 Fase de Conclusions i Memòria (Mes 3)

- Anàlisi dels resultats i avaluació d'una possible implementació real a nivell cost-benefici.
- Proposta d'implementació final del model en local.

5 ESTAT DE L'ART

Durant els últims anys la intel·ligència artificial ha estat protagonista en les tendències i ha portat molts canvis, consolidant-se en el sector empresarial on avui dia és vist com una eina molt útil, en menys de cinc anys, ha passat de ser una eina més a convertir-se en un pilar fonamental

per a qualsevol organització, en l'actualitat, els models LLM tenen el focus principal, ja que són sistemes basats en arquitectures de xarxes neuronals amb la capacitat de processar i generar text amb una coherència gairebé humana.

5.1 Fonaments dels LLM i Arquitectura Transformer

En l'actualitat, els models es basen majoritàriament en l'arquitectura Transformer [5], que permet processar les dades de manera paral·lela amb una millor comprensió del text.

En un principi aquest LLM requeria una capacitat de càlcul molt alta, únicament accessible per empreses amb superordinadors (HPC), però actualment, gràcies a tècniques de quantització (reduir la dimensió dels pesos dels models) [9] s'ha tornat un producte molt més accessible per l'usuari sense la necessitat de serveis al núvol, cosa que permet crear inferències locals.

5.2 Seguretat i Privacitat: Risc del cloud

Al mateix temps que s'optimitzava el producte, es va començar a conscienciar als usuaris i organitzacions del risc d'enviar informació privada a sistemes cloud públics, encara més perillós en el cas de propietats intel·lectuals.

En centres com el CNAG, l'alt risc de filtració o ús no autoritzat de dades genòmiques i privades impedeix utilitzar IA en entorns sense control. Això obliga a triar entre dues opcions: pagar llicències corporatives al núvol (amb costos des de 20-25 € per empleat, recursos limitats i el risc d'exposar codi intern) o crear sistemes d'IA On-Premise, que garanteixen el control absolut sobre la ubicació de les dades.

LLavors la situació inicial de l'entitat era de desavantatge, ja que, a causa del risc d'un possible data breach i la falta d'una alternativa local, la companyia no comptava amb aquesta eina, reduint la seva velocitat de desenvolupament en els seus projectes.

5.3 Seguretat i Privacitat: Risc en local

Malgrat garantir el control físic, la implementació de sistemes d'IA On-Premise trasllada tota la responsabilitat de la seguretat a l'organització. El risc principal en entorns locals es desplaça cap a les vulnerabilitats de la xarxa interna i la falta de governança de les dades.

En centres de recerca, l'ús de models locals pot provocar filtracions internes (data leakage) si usuaris no autoritzats extreuen informació confidencial o genòmica mitjançant consultes. A més, l'ús de programari de codi obert i la descàrrega de models externs introdueix el risc de patir atacs d'injecció de codi maliciós o portes del darrere (backdoors), exposant tota la propietat intel·lectual de l'entitat si la infraestructura local és vulnerada o no es manté actualitzada correctament.

5.4 Llicències Open Source

Gràcies a l'ecosistema del codi obert, han aparegut models amb rendiments comparables als tancats com GPT-4 on les seves llicències són gratuïtes, que permet que diferents organitzacions escollin el model més adient a les seves necessitats i/o equips disponibles, arribant al punt d'eines com Ollama [10] o vLLM [11] consolidar-se com l'estàndard per al desplegament correcte i eficient de models en entorns locals, tant a escala d'usuari com corporatiu.

És en aquest punt on CNAG busca la implementació d'aquests models per no perdre la seva posició dintre del seu sector juntament amb l'absència dels problemes dels models situats al cloud.

5.5 On és necessari la implementació?

A causa de no poder tenir un model d'IA tan avançat per pressupost, la idea és enfocar el model en una branca del centre en concret, en aquest cas, en programació, ja que, un model especialitzat en un únic treball podria funcionar a un nivell semblant dels models més generals tot i ser menys intel·ligents.

Avui dia moltes empreses no programen únicament en un sol llenguatge, han de tenir un front-end i back-end, model vista controlador,...

5.5.1 Que comporta aquesta situació?

El sistema resultant d'aquest treball ha de tenir la capacitat de programar en diferents llenguatges, això implica més varietat dels tests, en aquest cas, el centre ha comentat que vol tenir un model capaç de treballar amb JSON, SQL i Python.

En aquests casos, hi ha models preparats com Alibaba amb Qwen [7] en la branca coder, Meta amb CodeLlama o Deepseek amb el model Deepseek Coder.

Totes aquestes empreses ofereixen models LLM especialitzats en codi.

5.6 Els LLM engines

Perquè un model de llenguatge gran (LLM) sigui operatiu, es requereix un motor d'inferència (engine). Aquest programari és el responsable de carregar els pesos del model a la memòria i de gestionar el pipeline de còmput, actuant com a interfície entre l'estructura estàtica de la xarxa neuronal i l'aplicació final. Les seves funcions principals comprenen la tokenització, l'execució de la inferència i la gestió eficient de la memòria. Atès que un mateix model de llenguatge pot oferir taxes de transferència molt diverses segons el programari utilitzat, la selecció de l'orquestrador és crítica. Actualment, coexisteixen tres solucions dominants en l'estat de l'art:

- **Llama.cpp [12]**: Motor d'inferència de codi obert escrit en C++ natiu, dissenyat per oferir una alta portabilitat multiplataforma. Destaca per la seva capacitat d'executar tècniques de quantització basades en la reducció

de precisió dels pesos de la xarxa. Això permet fragmentar la càrrega de treball i fer un transvasament o descàrrega (offloading) de paràmetres cap a la memòria RAM del sistema, fent possible l'execució d'un model de 140 GB en targetes gràfiques limitades a 40 GB de VRAM en col·laboració amb la CPU.

- **Ollama**: Marc de treball (framework) orientat a l'usuari que utilitza llama.cpp com a motor de fons (backend). Opera en segon pla encarregant-se de l'automatització, la detecció de la GPU, l'enllaç de llibreries i la gestió del cicle de vida del model. Tot i que és l'estàndard per a entorns locals per la seva simplicitat en la configuració, sacrifica capacitat de rendiment brut a gran escala.
- **vLLM**: Motor de servei d'alta taxa de transferència (high-throughput) concebut específic de cara a entorns de producció i centres de dades concurrents. Dissenyat exclusivament per a arquitectures de GPU NVIDIA, incorpora l'algorisme PagedAttention. Aquesta funcionalitat simula la gestió de memòria virtual d'un sistema operatiu, paginant les dades del context per evitar la fragmentació de la VRAM. Actualment, es consolida com el motor de rendiment superior en entorns de concurrència massiva.

L'elecció del model està basat en els rànquings globals dintre de Hugging face, pàgina web on avaluen tota mena de models LLM i disposen de diferents rànquings globals basats en el tipus de tasca.

Avui dia també és el repositori dels models per poder descarregar-ho, convertint a Hugging face en la principal font d'informació, avaluació i obtenció de models LLM a escala global.

5.6.1 Quin és el millor teòricament?

Segons el rànking de Hugging face, el millor model seria MetaLlama 3.1 [6] per tasques generals i per programació Qwen Coder 2.5, per tant, la discussió estaria entre aquests 2 en un inici.

5.7 Els paràmetres dels LLM

Actualment, cada model LLM ofereix diferents variants pel que fa a paràmetres, que serien el nombre al costat del nom (4B, 8B, 32B 405B) això indica els bilions de paràmetres que té el model, o dit d'una altra manera, el nombre de connexions que hi ha dintre de la xarxa neuronal que forma el que entenem nosaltres com el cervell del model.

Com més paràmetres hi ha dintre del model, major memòria VRAM es necessita, però més intel·ligent és el model, que no ràpid, un nombre més elevat de connexions comporta un procés d'execució més llarg a l'hora d'utilitzar el model.

Aquests paràmetres actuen com les neurones del cervell humà, en el cas dels models LLM, contenen valors útils pel seu funcionament, en els models complets, sense cap classe de compressió de dades, l'estàndard és 2 bytes per model, però avui dia les empreses darrere dels models han ofert

versions comprimides dels models, on la grandària dels paràmetres està reduït a 4, 5 bits o 1 byte entre altres sense afectar significativament el rendiment del model.

Per tant, l'elecció dels models que s'utilitzaran respecte a paràmetres està definit per complet pel hardware disponible al centre, on el nombre de paràmetres haurà d'entrar dintre de la memòria VRAM del clúster o segment del clúster fet servir.

5.8 La bombolla de la IA

Actualment, moltes corporatives que ofereixen el seu model al cloud com Microsoft o X, tenen una relació entre cost i ingrés molt dispar, ja que, una subscripció pot costar entre 10 i 20 euros, però per a les corporatives hosts del sistema, cada tasca pot arribar a costar més de 20 \$ depenent del model, és a dir, avui dia, per l'augment de costos del hardware com la memòria RAM o les GPUs, només hi ha pèrdues en els sistemes d'IA del cloud.

Això suposa que els models d'IA del cloud no siguin rendibles corporativament i eventualment, els costos arribaran als usuaris, limitant recursos o augmentant els costos de les subscripcions, en els últims anys, les subscripcions han passat de respostes infinites a un límit de tokens mensuals per exemple.

Per tant, l'evolució demostra que el camí del futur és tenir models d'IA en local per tal de disminuir costos.

6 INVESTIGACIÓ INICIAL

6.1 Preguntes plantejades

Durant el primer mes d'investigació, hem trobat solució als dubtes que teníem respecte a la implementació d'un model d'IA, les preguntes inicials que vam tenir van ser:

- Quin hardware és millor, CPU o GPU? Podem considerar les NPUs?
- És millor tenir la IA implementada en cada ordinador o al clúster i que sigui per tothom?
- Quin hardware disposem o necessitarem?
- Com fer-ho de manera eficient en diners i consum?
- IA general o específica per a un treball en concret?
- És millor assumir el risc i utilitzar IA en línia?

6.1.1 Respostes obtingudes:

- **Arquitectura de computació (GPU vs. NPU):** Els LLM requereixen una gran densitat de VRAM i operacions de paral·lelització massiva de matrius, tasca òptima per a les GPU. Les NPU actuals estan limitades al mercat de consum i manquen de memòria dedicada per a models de gran escala.

- **Centralització del servei:** Desplegar el model centralitzat al clúster permet allotjar geometries de paràmetres més grans, millorant la correctesa i la velocitat. A més, redueix dràsticament els costos d'administració i manteniment de sistemes en unificar la infraestructura.
- **Criteri de selecció del maquinari:** El cost del maquinari HPC escala més ràpidament que el seu rendiment. L'estratègia s'ha de centrar en la relació cost-rendiment (performance-per-dollar), evitant components de gamma baixa que introdueixin latències crítiques en la xarxa.
- **Especialització del domini:** Per competir amb els models massius del Cloud, és imperatiu especialitzar el model local en un domini vertical concret que estigui en total sintonia amb els fluxos de treball del centre.
- **Sobirania i privacitat de les dades:** Atès que el CNAG gestiona actius crítics altament regulats (com seqüències d'ADN), l'ús de serveis externs al núvol introdueix un risc de fuga de dades (data breach) inadmissible per a l'entitat.

6.1.2 Quin és el producte resultant com a objectiu?

Una vegada tenim les respostes, podem definir quin producte final estem buscant:

- **Model centralitzat:** El model ha d'estar situat dintre del clúster per motius de rendiment, tant en temps com en qualitat de respostes, ja que permet major capacitat de còmput i, per tant, la capacitat de tenir models de més grandària de paràmetres, millorant la qualitat de les respostes i paral·lelament resolent els problemes de data breach.
- **Grandària del model:** Per òbvies raons de rendiment, el model seleccionat ha de poder situar-se al 100% a la memòria VRAM de la GPU que s'utilitzi, ja que, en cas contrari, pot minvar el temps d'obtenció de respostes.
- **Trobar la millor combinació:** S'ha de fer proves per veure en quina GPU s'obté els millors resultats pel que fa a qualitat i preu, pel fet que, l'augment de cost en infraestructura pot ser molt més significatiu que en millora de rendiment.
- **Model especialitzat:** Ja sigui a través d'un fine-tuning o per escollir models especialitzats en la tasca encomanada, s'ha d'obtenir un model expert en les tasques que es requereix la seva presència.

6.2 Clúster utilitzat i anàlisi del hardware

Per entendre com interactua el programari amb la infraestructura, cal considerar que un LLM funciona de manera similar a un cervell massiu amb una gran quantitat de dades internes. La mida d'aquest "cervell" es mesura mitjançant els paràmetres. Actualment, els models oscil·len des dels més bàsics (2, 4 o 8 milers de milions de paràmetres) fins als més avançats, que poden arribar al mig bilió.

Tanmateix, una quantitat més gran de paràmetres no sempre garanteix un millor resultat; l'eficiència del software i la precisió de les respostes estan estretament lligades a la

capacitat d'emmagatzematge en la VRAM (memòria de vídeo). Per trobar l'LLM més adequat per al CNAG, hem de buscar el model amb el nombre més elevat de paràmetres que pugui allotjar-se a la VRAM disponible. També podem considerar l'offloading, una tècnica que permet utilitzar la memòria RAM del sistema per donar suport a la GPU, tot i que això redueix la velocitat d'inferència a canvi de permetre models més grans.

6.2.1 Determinació de la mida i precisió:

La mida de cada paràmetre és variable: els més grans poden ocupar 2 bytes (FP16), mentre que els més petits, mitjançant tècniques de quantització, poden reduir-se a 4 o 5 bits. Cal tenir en compte que com més petit és el paràmetre, menor és la precisió del model. No ajustar correctament el model al hardware pot provocar una velocitat excessivament lenta o una infrautilització de les capacitats de la GPU.

6.2.2 Desglossament del hardware disponible al CNAG:

El centre disposa de diferents configuracions que requereixen aproximacions específiques:

- **Grup 1: NVIDIA H200:** Disposa de 8 unitats NVIDIA H200 amb un total de 1152 GB de VRAM, 128 nuclis de CPU i 2 TB de RAM. És el hardware ideal per a models de nivell alt (centenars de milers de milions de paràmetres com Llama 3 405B o DeepSeek-V3) amb una velocitat excel·lent. Tot i això, la seva disponibilitat per a aquest projecte és molt limitada.
- **Grup 2: NVIDIA L40:** Compta amb 2 GPUs L40 amb un total de 48 GB de VRAM, 64 nuclis de CPU i 1 TB de RAM per ordinador. El model més adequat aquí seria un de **70B paràmetres** quantitzat a 4 o 5 bits, o un de **32B** amb una precisió superior.
- **Workstations RTX A6000:** Dues estacions de treball amb un total de 48 GB de VRAM i 128 GB de RAM per ordinador. En aquest cas, un model de **32B** és més recomanable, ja que la limitada RAM del sistema dificulta l'estratègia d'offloading si fos necessària.
- **Workstation NVIDIA RTX 6000:** Una estació amb dues targetes que sumen 48 GB de VRAM i 128 GB de RAM, presentant limitacions de capacitat similars a les anteriors.

6.3 Com saber el nombre de paràmetres optims

Un cop determinat l'interval de paràmetres, cal avaluar el compromís entre una menor quantitat de paràmetres amb alta precisió (32B a 8-bit) o una densitat més gran de paràmetres amb una compressió més alta (70B a 4-bit). La decisió s'ha de fonamentar estrictament en la naturalesa de les tasques del centre:

- **Avantatges de la densitat més elevada (70B):** Per a tasques bases en raonament complex, comprensió lectora i resums, el fet de disposar de més del doble de connexions neuronals aporta un avantatge cognitiu superior. Aquesta densitat compensa la pèrdua de precisió derivada d'una quantització més estricta.
- **Avantatges de la precisió més alta (32B):** Per a aplicacions que requereixen una alta precisió matemàtica o

rigor tècnic extrem, conservar les dimensions originals dels pesos evita la pèrdua d'informació microscòpica, superant els models més densos però altament complexos.

Atesos els requisits del CNAG, enfocats principalment en la generació de codi i l'extracció de dades estructurades, inicialment es podria preveure un millor rendiment en el model de 32B. No obstant això, en l'estat de l'art actual, la degradació que pateixen els models de 70B quantitzats a 4 bits en codis complexos és pràcticament imperceptible. En línies generals, l'arquitectura de 70B es consolida com una solució més completa i robusta per a l'entitat.

6.4 Conclusió obtinguda: Selecció del model i arquitectura

Una vegada analitzat el hardware disponible i les capacitats dels LLM segons la seva mida i quantització, podem definir l'estratègia d'implementació. Cal destacar que, per poder realitzar comparatives vàlides entre GPUs, necessitem que el model estigui optimitzat per a les arquitectures corresponents: NVIDIA (CUDA/Harper).

Requisits dels models		
Model LLM	RAM necessària pel Model	Ram per la cache Key Value
MetaLlama 3.1 70B instruct.	40 GB	8 GB en L40, A6000 i les RTX 6000 100 GB en la H200
Qwen 2.5 Coder 32B	36 GB	12 GB en L40, A6000 i les RTX 6000 104 GB en la H200
CodeLlama 70B	40 GB	8 GB en L40, A6000 i les RTX 6000 100 GB en la H200

Figura 1. Taula de requisits de VRAM

7 PROVES

7.1 Context de l'execució de proves

7.1.1 Entorn d'execució de proves

Per poder dur a terme les proves descrites a la metodologia, s'ha configurat un entorn de treball amb propòsit experimental dintre del centre, complint els estàndards de l'enginyeria de sistemes.

Tota la interacció amb el clúster i els frameworks d'IA és gestionat a través d'una terminal Bash, l'ús d'aquesta terminal permet tenir alguns avantatges, podem utilitzar eines com "nvidia-smi" o "htop" per poder monitorar diferents dades com el consum de recursos de RAM o CPU si cal, addicionalment, agilitzem les proves gràcies a poder llençar la prova de manera automatitzada mitjançant scripts i fer les proves de rendiment o velocitat de manera molt més ràpida, finalment i també més important, mantenim la seguretat del centre gràcies a la connexió SSH amb claus digitals.

7.1.2 Factors a avaluar

A causa de l'elevat nombre de variables interdependents en un sistema d'IA On-Premise, s'aplica una estratègia experimental basada en l'aïllament de variables: s'executaran assajos comparatius individuals per a cada dimensió, mantenint els altres components constants.

L'arquitectura es desglossa en tres dimensions crítiques d'avaluació:

1. **Motor d'Inferència (LLM Engine):** S'avalua l'eficiència de la gestió de memòria i la concurrència (vLLM, llama.cpp, Ollama).
2. **Model de Llenguatge:** S'analitza quin és el model òptim optimitzat per a la restricció física del maquinari seleccionat com a host. Es compara el rendiment i qualitat de respostes entre models de densitat i paràmetres similars.
3. **Infraestructura de Maquinari (Hardware):** S'avalua el rendiment segons l'entorn d'execució, analitzant quina de les GPUs disponibles en conjunt o en solitari donen millor resultat.

7.2 Proves efectuades

El mes següent, per avaluar el nostre model, compararem el seu rendiment amb models coneguts disponibles en línia en diferents punts clau:

- Intervenció humana
- Correctesa
- Eficiència
- Recursos utilitzats
- Temps

Tot i ser factors a tenir en compte, alguns tenen prioritat per sobre d'altres, aquí està l'ordre en el qual m'he basat per avaluar els resultats, tot i que no és immutable:

1. **Correctesa:** A part de valorar si és correcta o no, hi ha respostes amb millor solució que altres, per exemple, si es demana reduir la complexitat de N^2 , i s'obté una reducció a N i altre a 1, ambdues solucions són correctes, però la segona és una resposta de millor qualitat perquè la reducció és major.
2. **Temps:** En cas d'empat a correctesa, el següent punt més valorat és el temps, on un menor temps en obtenir la resposta significa una millor posició.
3. **Intervenció Humana:** Tot i obtenir una bona solució i un bon temps, si s'ha requerit més intervencions humanes o les intervencions humanes requerides han significat molta informació extra al prompt, disminueix l'aprovació de les respostes.
4. **Eficiència i recursos utilitzats:** En aquest cas, situem ambdues opcions com iguals, en una valorem la qualitat del codi, donant millors respostes des d'un punt de vista computacional i en l'altre es valora en quin equip fa servir el model, tant en recursos de hardware com de consum si s'escau.

7.2.1 Segmentació de les proves

Per a l'execució dels assajos, s'han dissenyat quatre fitxers executables parametritzats. Aquesta arquitectura permet

commutar dinàmicament entre models, GPUs i motors d'inferència mitjançant variables de configuració inicials, garantint la immutabilitat dels prompts i la reproductibilitat del benchmark.

- **Test 1: Extracció de dades estructurades (Data Extraction):** Processament en lot (batching) de 20 registres de logs del sistema. L'objectiu és l'extracció sintàctica cap a un esquema JSON vàlid que classifiqui variables com el timestamp, nivell de severitat i fitxer d'origen. S'avalua el temps d'execució, la integritat del format i la taxa d'encert.
- **Test 2: Generació de tests unitaris (Unit Testing):** S'entrega una funció en Python d'alta complexitat amb l'objectiu d'avaluar la capacitat de raonament del model en l'assegurament de programari. S'exigeix el disseny de casos extrems (l·listes buides, tipus invàlids o claus absents). S'analitza la correctesa, la cobertura i la qualitat del codi resultant.
- **Test 3: Disseny de consultes estructurades (SQL Queries):** Traducció de llenguatge natural a codi PostgreSQL complex a partir d'un esquema relacional de bases de dades biomèdiques. S'avalua la correcció sintàctica, l'eficiència de les unions (joins) i la precisió lògica de la consulta.
- **Test 4: Optimització de codi i refactorització (HPC):** Refactorització d'un algoritme ineficient amb complexitat temporal neta de $O(N^M)$ cap a un enfocament vectoritzat d'alt rendiment amb NumPy. S'avalua la reducció de la complexitat asimptòtica, la viabilitat d'execució i la capacitat d'explicació tècnica del model.

Així mateix, transversalment es computa la mètrica d'Intervenció Humana (IH). Aquest paràmetre quantifica el nombre de correccions o indicacions addicionals inserides al context en cas que el model inicialment gènere codi erroni o requereixi guies de relació per assolir una solució funcional.

La quantitat de tokens d'entrada i sortida estan annexats al punt A1 i A2, i el codi que conté els tests està annexat al A15.

7.2.2 Proves efectuades pels LLM engine

Per determinar el motor, es va executar el test de temps més (alt), el test 1 que de mitjana ha estat el 53,24% del temps total d'execució, reduint la seva complexitat, però mantenint el volum, d'aquesta manera, accentuem més la diferència de velocitats entre motors LLM per decidir el motor LLM per la resta de proves.

En concret, les condicions han estat:

- Hardware: NVIDIA L40, ja que, vLLM no suporta gràfiques AMD.
- Model LLM utilitzat: Llama 3.1 70B instruct format GGUF.
- Test modificats per accentuar la velocitat en tokens/s.

7.3 Resultats obtinguts

Els resultats obtinguts són per les següents proves efectuades:

- Proves per determinar el millor LLM engine.

- Proves per determinar el millor LLM model en cada GPU.
- Proves per determinar la millor GPU.

7.3.1 Resultats obtinguts dels LLM engines

Amb les proves s'ha obtingut els següents resultats:

- Llama.cpp: Rendiment desequilibrat tot i tenir una velocitat d'entrada de 641,2 tokens/s, la velocitat de 16,8 tokens/s de sortida/generació ha estat la limitant amb diferència, necessitant una mitjana de 39,84 s en resol-dre.
- Ollama: El rendiment no és el seu punt fort, també cal destacar que Ollama ha de descarregar el model i guardar-ho en una estructura basada en imatges de docker, que pot arribar a afectar en el rendiment negativament, la seva velocitat ha estat de 4,95 tokens/s, acabant la prova en 135,21 s.
- vLLM: Gràcies a la seva tècnica de paginació, redueix el temps d'accés a les dades, molt notori en els 157,78 tokens/s de velocitat, obtenint la solució en 4,69 s.

7.3.2 Avaluació d'infraestructura i viabilitat de vLLM al centre

L'adopció de vLLM s'ha d'avaluar equilibrant les necessitats de rendiment de l'entitat amb les seves restriccions de maquinari. Per al CNAG, disposar d'un model local especialitzat és un factor estratègic i competitiu essencial. No obstant això, el valor d'aquesta solució depèn directament de la seva velocitat de processament i generació (Throughput).

El principal inconvenient de vLLM és la seva incompatibilitat amb el programari d'AMD [8], plataforma que el centre considerava inicialment com a host, però que es va descartar durant la investigació per problemes de drivers. Malgrat aquesta limitació, vLLM i l'ecosistema NVIDIA (CUDA) s'han consolidat com l'estàndard industrial de facto en entorns corporatius i de computació d'alt rendiment (HPC).

Considerant el maquinari disponible al clúster, vLLM és l'opció més adequada per al projecte. L'optimització de la seva latència garanteix un servei multiusuari fluid i un estalvi net de recursos computacionals. A més, aquesta elecció assegura la línia d'escalabilitat del centre, que disposa d'infraestructura NVIDIA addicional per allotjar models de densitat més alta de paràmetres en el futur.

7.4 Resultats de la H200

Abans de començar comentar que les proves amb la H200 es van fer al final, a aquest punt vaig decidir descartar CodeLlama 70B pels resultats obtinguts, com a substitució, vaig afegir MetaLlama 3.1 Instruct 70B pero descomprimit, per veure si hi havia diferència de respostes entre un model descomprimit i comprimit en les proves fetes.

Els resultats de les proves estan annexats als punts A9 i A10, on es recull el percentatge de encert i temps d'execució.

7.4.1 Comparació cost-rendiment

Un altre punt a destacar és la millora de rendiment i la diferència de costos. En un principi, en el món tecnològic, pagar el doble no significa tenir el doble de rendiment, cada vegada més, l'augment de rendiment és menor a l'augment de cost, però en aquest cas no es compleix.

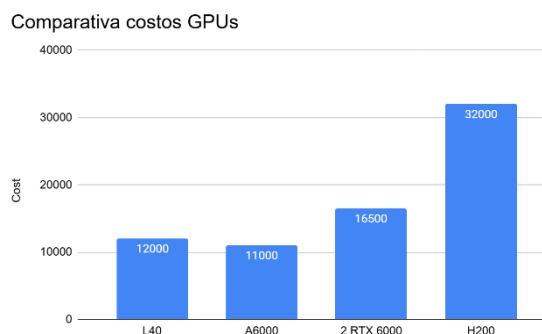


Figura 2. Gràfic comparatiu de preus de GPU

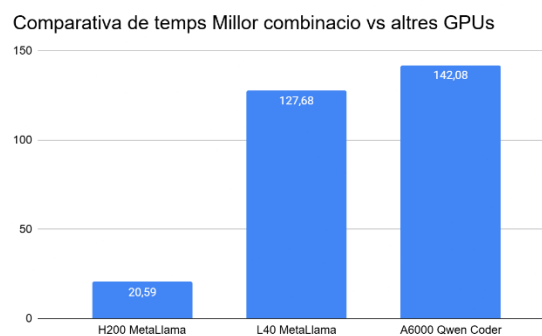


Figura 3. Gràfic comparatiu de temps de H200 vs millors combinacions

7.4.2 Resultats entre comprimit i descomprimit

Per últim, cal comentar el fet que el model MetaLlama 3.1 comprimit no només ha obtingut els mateixos resultats que el model descomprimit, sinó que a més a més ha tingut un millor rendiment en temps.

Els resultats de les proves els teniu disponibles al final del document, a una taula annexada.

7.4.3 Millor model LLM a cada test.

El test 1 des de l'inici ha estat dominat per MetaLlama 3.1 i en aquest cas no ha canviat res, ha demostrat una velocitat molt alta en ambdues versions, sent aproximadament 4 vegades més ràpid que Qwen, entre les dues versions, la descomprimida ha estat la més ràpida.

Al test 2, per correctesa ha estat millor per Qwen, de nou, s'ha necessitat una intervenció humana per MetaLlama 3.1.

En el test 3, Qwen de nou ha estat millor, en aquest cas per velocitat.

Per últim, tot i ser més ràpid Qwen al test 4, per correctesa MetaLlama ha donat una resposta de millor qualitat en ambdues versions, tot i que, de nou la versió comprimita ha tingut lleugerament millor velocitat.

7.5 Taula resultats

A causa de la dimensió de les proves, els resultats estan annexats des del punt A3 fins al A14, recollint el temps per execució i percentatge d'encert, en cas de voler saber la resta de dades recopilades, el annex A16 dona accés a un GitHub amb tots els resultats obtinguts.

7.6. Anàlisi dels resultats sense comptar H200

7.6.1 Correctesa dels resultats

Qwen-coder tot i tenir respostes correctes, l'últim test no ha donat la millor resposta possible, utilitzava `numpy.newaxis` quan la solució òptima era amb `numpy.linalg.norm`, reduint nombre de variables i fent servir funcions de la galeria C òptimes.

D'igual manera, Meta Llama amb la A6000 ha donat una resposta on no ha reduït del tot la complexitat del codi inicial, complint amb l'optimització, però no donant la millor solució possible.

En el tercer test, també s'ha valorat, algunes respostes utilitzaven distint i group by amb les mateixes variables, donant com a resultat una operació innecessària extra, minvant el temps d'execució de la comanda en un cas real.

D'altra banda, comentar que a causa de la falta de correctesa del model CodeLlama 70B i el gran cost per al centre de deixar-me la Nvidia H200 pels experiments, vaig decidir descartar aquest model i fer una altra prova, on compararia el model MetaLlama 3.1 instruct de 70B descomprimit per veure si hi ha alguna diferència de rendiment o correctesa.

7.6.2 Importància de les Intervencions humanes

Tot i comptar igual, hi ha intervencions humanes més grans que altres, tenint en compte la quantitat d'informació afegida al prompt, o el seu impacte al resultat, he valorat quina intervenció té més pes.

7.6.3 Millor resultat de cada test

Valorant tots els factors avaluats, els millors resultats per cada test son:

- Test 1 Extreure log a JSON: Meta Llama 3.1 amb Nvidia L40, en aquest test, la resposta era binària, és a dir, correcta o incorrecta, llavors el factor principal seria el temps necessari.
- Test 2 Millorar funció Python: En aquest test, quan la resposta és 100% correcta, vol dir que ha tingut en compte totes les possibilitats possibles, com tenir llistes buides, variables incompatibles entre altres, el millor resultat ha estat Qwen Coder amb les dues Nvidia RTX 6000.
- Test 3 SQL Query: Com s'ha comentat amb anterioritat entre les respostes amb 100% d'encert hi ha diferències de correctesa, en aquest cas, de nou les dues Nvidia RTX 6000 amb Qwen Coder han estat el millor resultat.
- Test 4 Optimització amb numpy: De nou, hi ha millors solucions que altres, per tant, entre les encertades, la

millor de totes ha estat la Nvidia L40 amb Meta Llama 3.1 com a model.

8 CONCLUSIÓ

8.1 Sistema IA local final

Una vegada avaluat punt per punt els elements que formen el model, el sistema final és el següent:

- vLLM com a motor LLM.
- Nvidia H200 com a GPU host del model.
- MetaLlama 3.1 instruct 70B comprimit o Qwen coder 2.5 com a model LLM.

8.2 Conclusions obtingudes dels LLM engines

La diferència en rendiment que ofereix vLLM respecte als seus competidors, és més que suficient per compensar la seva falta de compatibilitat amb AMD.

8.3 Selecció de la GPU

En aquest cas la resposta inicials és fàcil, la H200, però traient la H200? Considerant els resultats, per temps i per correctesa la millor ha estat la L40, seguit a prop per la RTX A6000 i, en últim lloc les RTX 6000.

8.4 Selecció del model LLM

CodeLlama 70B es descarta de forma definitiva a causa d'una taxa de correctesa insuficient en els assajos inicials. Per a la decisió final del CNAG es plantegen dues alternatives: Meta-Llama 3.1 Instruct 70B, que lidera el rendiment temporal brut (especialment en el Test 1 i sota nodes H200 o L40), i Qwen 2.5 Coder 32B, que destaca per una consistència estructural superior amb prop d'un 100% d'encert transversal. S'aconsella Llama 3.1 si es prioritza la velocitat pura en els nodes de computació principals, o Qwen 32B si es busca flexibilitat i estabilitat per desplegar el sistema de manera distribuïda en qualsevol GPU disponible del clúster.

8.5 Comparació amb els models del Cloud

Des del punt A11 fins al A14 del annex tenim els resultats i comparatives entre el model definitiu resultant del treball i el cloud, amb les dades recollides podem confirmar que el nostre model és millor com a mínim 0,5 vegades més que el millor temps obtingut al cloud.

Per saber si és realment rendible, cal comentar que en aquest cas, independentment del fet que el model sigui implementat o no, pel CNAG els costos d'electricitat i personal de manteniment són iguals, ja que, el clúster utilitza de manera activa aquest equip, únicament seria un canvi d'activitat associada a aquell equip en concret.

Així i tot, el consum mig de l'equip és aproximadament 1200 W, és a dir, amb els costos d'electricitat actuals a Espanya, el cost seria de 160 euros/mes si està encès constantment o 55 euros en el cas d'estar encès 8 h/dia.

Això significa que en cas de tenir més de 3 usuaris connectats, el cost seria inferior a les subscripcions del cloud.

8.6 Replicabilitat de l'experiment i guia de desplegament corporatiu

Aquesta metodologia és un marc de treball reproducible per desplegar IA local en qualsevol organització mitjançant un protocol de quatre passos: el **dimensionament del workload** per definir la mida del model segons les tasques; l'**auditoria del maquinari** per triar la infraestructura (prioritzant NVIDIA per a alt rendiment); el **disseny del banc de proves** per mesurar sistemàticament la velocitat i correctesa del sistema; i la **viabilitat econòmica**, que pondera el cost del *hardware* (optimitzat amb tècniques de compressió) enfront dels beneficis de privacitat i sobirania del coneixement.

ÚS DE LA IA GENERATIVA

Principalment, la IA generativa s'ha utilitzat per traduir de l'anglès els documents que he fet pel CNAG i per professionalitzar o resumir el text de l'informe, addicionalment s'ha fet servir per a l'enteniment de conceptes relacionats amb el treball però en casos excepcionals en aquest cas.

AGRAÏMENTS

Vull agrair principalment al CNAG per donar-me l'oportunitat d'accedir a un hardware tant privilegiat, a FIATC Assegurances pels suggeriments rebut en analitzar el treball per aplicar-ho ells mateixos, a la meua família pel suport i a l'Antoni Espinosa per guiar-me en tenir un producte de millor qualitat.

BIBLIOGRAFIA

- [1] Zendesk, "¿Qué es un clúster y para qué sirve en las empresas?" Zendesk. Consultat el 1 de març de 2026. [En línia]. Disponible a: <https://www.zendesk.com.mx/blog/cluster-que-es/>
- [2] Wikipedia, "Violación de datos," Wikipedia, la enciclopedia libre. Consultat el 1 de març de 2026. [En línia]. Disponible a: https://es.wikipedia.org/wiki/Violación_de_datos
- [3] CNAG, "Centre Nacional d'Anàlisi Genòmic - Home," CNAG. Consultat el 1 de març de 2026. [En línia]. Disponible a: <https://www.cnag.eu/>
- [4] Amazon Web Services (AWS), "¿Qué es un modelo de lenguaje de gran tamaño (LLM)?" AWS. Consultat el 10 d'abril de 2026. [En línia]. Disponible a: <https://aws.amazon.com/es/what-is/large-language-model/>
- [5] A. Vaswani et al., "Attention Is All You Need," arXiv (Google Research). El paper original que defineix la arquitectura **Transformer**. Consultat el 10 d'abril de 2026. [En línia]. Disponible a: <https://arxiv.org/abs/1706.03762>
- [6] Meta AI, "Introducing Llama 3.1: Our most capable open models to date," Meta. Consultat el 10 d'abril de 2026. [En línia]. Disponible a: <https://ai.meta.com/blog/meta-llama-3-1/>
- [7] Qwen Team, "Qwen2.5-Coder: Toward the Open-Source Era of Code LLMs," GitHub/Hugging Face. Consultat el 10 d'abril de 2026. [En línia]. Disponible a: <https://qwenlm.github.io/blog/qwen2.5-coder/>
- [8] AMD, "ROCm Software Platform for AI and HPC," AMD. Explicació de l'arquitectura per a GPUs Radeon. Consultat el 10 d'abril de

2026. [En línia]. Disponible a: <https://www.amd.com/en/developer/rocm.html>

[9] T. Dettmers et al., "QLoRA: Efficient Finetuning of Quantized LLMs," arXiv. Referència fonamental sobre la **quantització** i com reduir la precisió dels pesos (bits). Consultat el 10 d'abril de 2026. [En línia]. Disponible a: <https://arxiv.org/abs/2305.14314>

[10] Ollama, "Ollama Documentation: Run LLMs locally," Ollama. Consultat el 10 d'abril de 2026. [En línia]. Disponible a: <https://ollama.com/library>

[11] vLLM Project Team, «vLLM: Easy, Fast, and Cheap LLM Serving with PagedAttention and Virtual Memory Management», *vLLM Architecture Foundation*, 14 de juny de 2026. [En línia]. Disponible a: <https://github.com/vllm-project/vllm>

[12] G. Gerganov, «Llama.cpp: Inference of Llama model in pure C/C++ without external dependencies», *GitHub Repository*, 14 de juny de 2026. [En línia]. Disponible a: <https://github.com/ggerganov/llama.cpp>

APÈNDIX

A1. Gràfic 1: Input de tokens per test

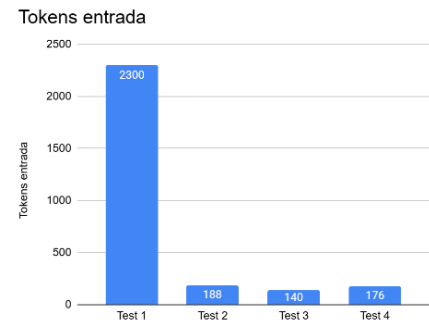


Figura 4. Gràfic que recull els tokens d'entrada als tests

A2. Gràfic 2: Output de tokens per test

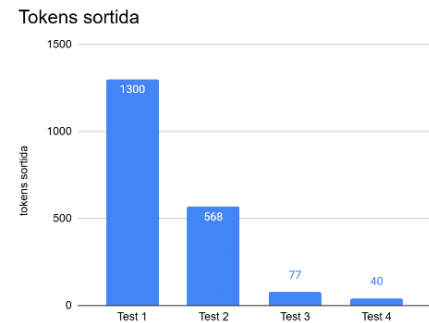


Figura 5. Gràfic que recull els tokens de sortida als tests

A3. Gàfica 3: Temps d'execució L40

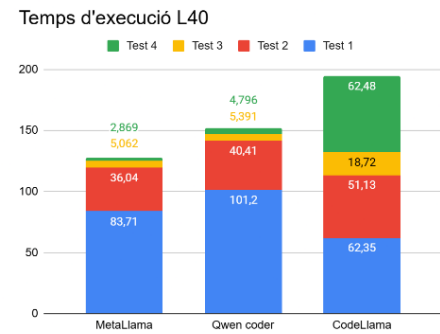


Figura 6. Gràfic que recull els temps d'execució amb L40

A4. Gàfica 4: Percentatge Encert L40

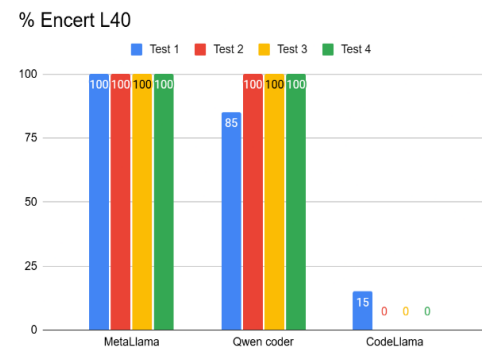


Figura 7. Gràfic que recull el percentatge d'encert amb L40

A5. Gàfica 5: Temps d'execució A6000

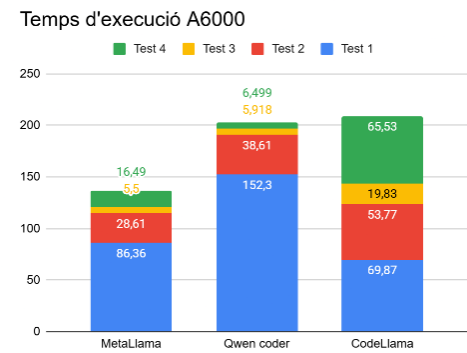


Figura 8. Gràfic que recull els temps d'execució amb la A6000

A6. Gàfica 6: Percentatge Encert A6000

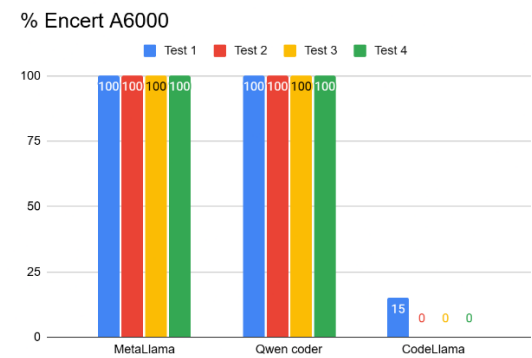


Figura 9. Gràfic que recull els percentatges d'encert amb la A6000

A7. Gàfica 7: Temps d'execució 2 RTX 6000

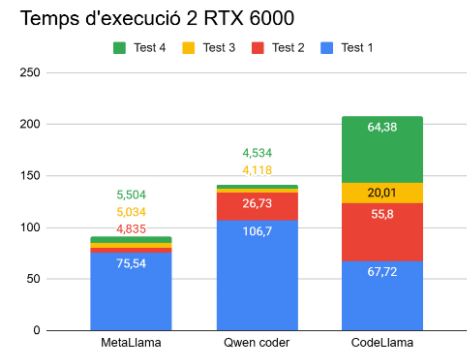


Figura 10. Gràfic que recull el temps d'execució amb les RTX 6000

A8. Gàfica 8: Percentatge Encert 2 RTX 6000

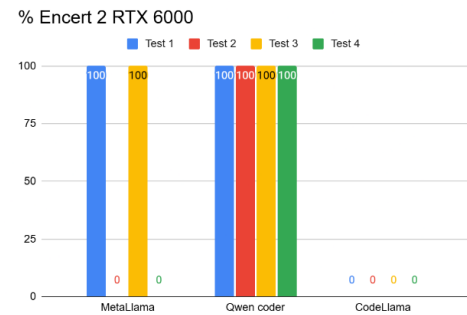


Figura 11. Gràfic que recull el percentatge d'encert amb les RTX 6000

A9. Gàfica 9: Temps d'execució H200

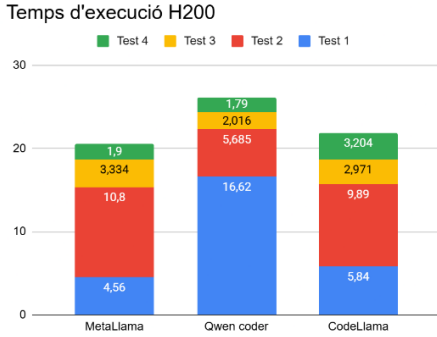


Figura 12. Gràfic que recull el temps d'execució amb la H200

A10. Gàfica 10: Percentatge Encert H200

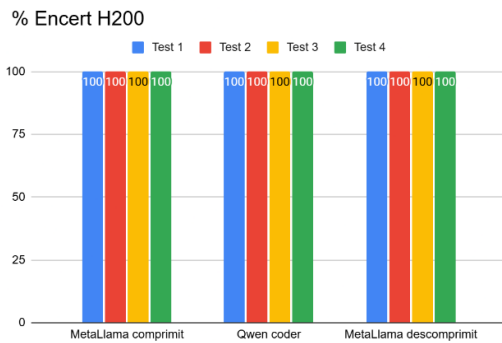


Figura 13. Gràfic que recull els percentatges d'encert amb la H200

A11. Gàfica 11: Temps d'execució models cloud

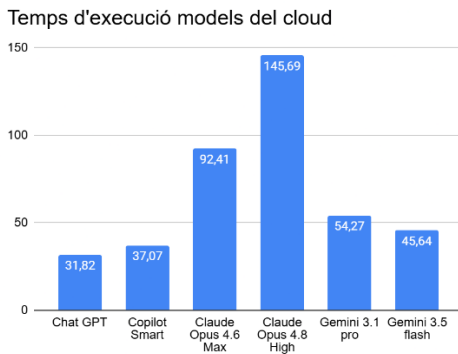


Figura 14. Gràfic que recull els temps d'execució dels models del cloud

A12. Gàfica 12: Percentatge Encert cloud 1

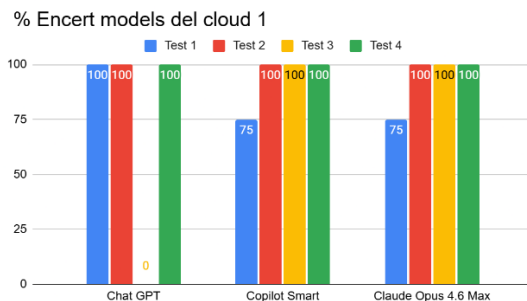


Figura 15. Gràfic que recull els percentatges d'encert dels models del cloud 1

A13. Gàfica 13: Percentatge Encert cloud 2

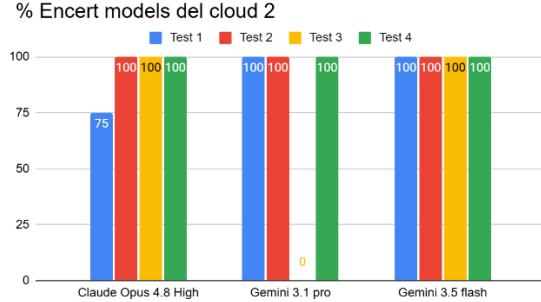


Figura 16. Gràfic que recull els percentatges d'encert dels models del cloud 2

A14. Gàfica 15: Comparativa de temps entre cloud i millor combinació

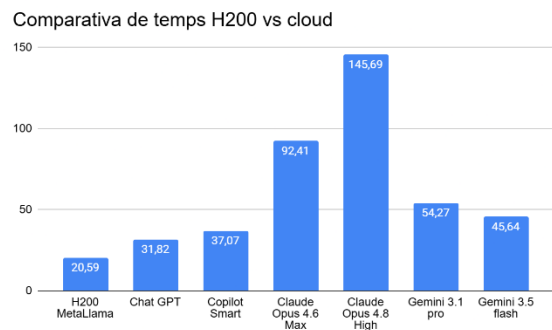


Figura 17. Gràfic comparatiu entre cloud i la millor combinació local

A15. Link als prompts utilitzats a les proves

Enllaç al script amb els prompts de les proves:

https://github.com/NIU1665775/TFG_CNAG/blob/main/script.sh

A16. Github amb resultats complet de les proves

Enllaç al repositori amb el resultat de les proves:

https://github.com/NIU1665775/TFG_CNAG.git