Compressió de dades amb tecnologia de GPU

Eric Chaves Sánchez

Resum—Aquest treball presenta una comparativa entre diferents algorismes de compressió aplicats a dades científiques provinents del Sincrotró ALBA. Es contrasten solucions tradicionals basades en CPU amb implementacions modernes accelerades per GPU mitjançant CUDA. L'anàlisi es basa en un conjunt de 1800 imatges (65,3 GB), mesurant la ràtio de compressió, temps i eficiència. Els resultats mostren les diferències en rendiment segons la tecnologia utilitzada. Això permet orientar la selecció d'algorismes segons els requisits d'emmagatzematge i temps real.

Paraules clau—Anàlisi comparativa, BZIP2, Compressió de dades, CUDA, GPU, LZ4, Rendiment computacional, Sincrotró ALBA.

Abstract—This work presents a comparison of several compression algorithms applied to scientific data from the ALBA Synchrotron. It contrasts traditional CPU-based solutions with modern GPU-accelerated implementations using CUDA. The evaluation uses a dataset of 1800 images (65.3 GB), measuring compression ratio, times, and efficiency. Results reveal clear differences in performance depending on the technology. This guides the selection of algorithms based on storage and real-time processing needs.

Index Terms—ALBA Synchrotron, BZIP2, Comparative analysis, CUDA, Data compression, GPU, LZ4,Computational performance.

1 INTRODUCCIÓ

En el panorama científic actual, les grans instal·lacions experimentals, com ara sincrotrons i acceleradors de partícules, s'han convertit en fonts massives de dades. La capacitat de generar volums d'informació a velocitats sense precedents planteja reptes tecnològics de primer ordre, especialment en l'àmbit de l'emmagatzematge, la transmissió i l'anàlisi de resultats en temps útil.

El Sincrotró ALBA és un exemple clar d'aquest context, on els experiments generen un flux de dades científiques a una velocitat molt elevada. Aquesta allau d'informació, sovint en forma d'imatges de detectors d'alta resolució, requereix solucions eficients que permetin gestionar-la de manera pràctica. La necessitat de processar aquestes dades en temps real o quasi real sense comprometre la integritat de la informació científica és un dels principals motors d'aquest projecte.

Una de les estratègies fonamentals per afrontar aquest repte és la compressió de dades. Aquest treball se centra no només en l'aplicació d'algorismes de compressió, sinó en l'exploració del potencial que ofereix la tecnologia de les Unitats de Processament Gràfic (GPU) per accelerar aquests processos. A diferència de les CPU tradicionals, les GPU estan dissenyades per a la computació massivament paral·lela, una característica que pot ser aprofitada per a optimitzar tasques de compressió. L'estudi aprofundeix en l'ús de la plataforma CUDA de NVIDIA com a eina per desenvolupar i executar solucions de compressió d'alt rendiment.

L'objectiu principal d'aquest Treball de Fi de Grau és, per tant, realitzar una anàlisi comparativa exhaustiva entre diferents eines i algorismes de compressió, tant executats en CPU com optimitzats per a GPU. S'avalua el rendiment de solucions clàssiques com LZ4 i BZIP2 al costat d'implementacions específiques per a CUDA, com CUDA_BSC o CUDA-Image-and-Video-Codec. L'avaluació es basa en un conjunt de mètriques quantitatives rigoroses, com els temps de compressió i descompressió, la ràtio de compressió assolida i la taxa de processament , amb la finalitat de determinar quines solucions ofereixen el millor equilibri per a les necessitats operatives del Sincrotró ALBA.

Per assolir aquests objectius, la memòria s'estructura en els següents capítols. En primer lloc, es detallen els objectius específics del treball. A continuació, es presenta el marc teòric que descriu els algorismes estudiats i l'entorn experimental. Posteriorment, s'exposa la metodologia seguida i s'analitzen en detall els resultats obtinguts. Finalment, es discuteix la viabilitat de cada solució en el context d'ALBA, es presenten

- *E-mail de contacte: 1633944@uab.cat*
- *Menció realitzada: Tecnologies de la Informació*
- *Treball tutoritzat per: Joan Bartrina Rapesta (DEIC)*
- *Curs 2024/25*

les conclusions generals i les possibles línies de treball futur.

2. OBJECTIUS

L'objectiu principal d'aquest Treball de Fi de Grau és realitzar una anàlisi comparativa exhaustiva d'algorismes de compressió en CPU i GPU, aplicats a les dades científiques del Sincrotró ALBA. La finalitat és determinar quina solució ofereix el millor equilibri entre velocitat, eficiència i qualitat de les dades per a un entorn que requereix un alt rendiment.

Per aconseguir-ho, s'han marcat els següents objectius específics:

- Analitzar les diferències de rendiment entre algorismes clàssics de CPU i moderns de GPU (CUDA).
- Avaluar el grau de compressió, els temps de procés i l'impacte en la qualitat de les dades científiques.
- Documentar les dificultats pràctiques d'instal·lació i configuració dels algorismes en el maquinari disponible.
- Crear una metodologia d'avaluació clara i replicable per a futurs estudis similars.
- Proporcionar recomanacions tècniques pràctiques i aplicables per al Sincrotró ALBA.

3. MARC TEÒRIC

Aquest capítol estableix les bases teòriques necessàries per comprendre el treball realitzat. En primer lloc, s'introdueixen els conceptes clau de la compressió de dades i la computació amb GPU. Posteriorment, es descriuen en detall les característiques i el funcionament de cadascun dels algorismes de compressió i descompressió que han estat seleccionats i avaluats en aquest projecte.

3.1 Fonaments de la compressió de dades

La compressió de dades és el procés de codificar informació utilitzant menys bits que la representació original. L'objectiu principal és reduir la redundància en les dades per minimitzar-ne la mida, facilitant-ne l'emmagatzematge i la transmissió. Existeixen dues grans categories:

- **Compressió sense pèrdua (Lossless):** Redueix la mida del fitxer sense perdre cap informació. En descomprimir, es recupera el fitxer original de manera exacta.
- **Compressió amb pèrdua (Lossy):** Elimina informació considerada menys important per aconseguir ratios de compressió més elevats. El fitxer descomprimit és una aproximació de l'original.

3.2 Compressió GPU amb CUDA

Les Unitats de Processament Gràfic (GPU) són dispositius dissenyats per al processament massivament paral·lel. A diferència de les CPU, que tenen pocs nuclis optimitzats per a tasques seqüencials, les GPU disposen de milers de nuclis més senzills, ideals per executar la mateixa operació sobre grans conjunts de dades simultàniament.

CUDA (Compute Unified Device Architecture) és la plataforma de computació paral·lela i model de programació desenvolupat per NVIDIA. Permet als desenvolupadors utilitzar la potència de les GPU per a tasques de computació de propòsit general, accelerant significativament aplicacions en camps com la ciència, l'enginyeria i, com és el cas d'aquest treball, la compressió de dades.

3.3 Algorismes de compressió

A continuació, es detallen els algorismes de compressió i descompressió utilitzats en el projecte.

3.3.1 CUDA_BSC.

CUDA_BSC és una implementació de compressió per a fluxos de dades binàries de baixa entropia. Utilitza una potent canonada de tres etapes: primer, la transformada de Burrows-Wheeler (BWT) per agrupar patrons; després, una codificació Run-Length (RLE) per a seqüències repetides; i finalment, una codificació d'Huffman. Tot el procés està dissenyat per a una execució massivament paral·lela a la GPU mitjançant CUDA, buscant una alta eficiència tant en la compressió com en la descompressió.

3.3.2 CUDA_ICPP.

Aquesta eina no és un algorisme per si mateix, sinó un entorn de treball per integrar diverses tècniques de compressió (com LZ77 o BWT) directament a la GPU. El seu objectiu principal és reduir el costós trànsit de dades entre la CPU i la GPU, comprimint la informació allà on ja s'està processant. Això el fa ideal per a fluxos de treball on les dades ja resideixen a la memòria de la GPU per a altres tasques computacionals.

3.3.3 CUDA-LZSS-UNKNOWN

CUDA-LZSS-UNKNOWN és una implementació en CUDA té una funció molt específica: està dissenyada exclusivament per a la descompressió de dades. Es basa en l'algorisme LZSS, reconstruint el fitxer original a partir de referències a dades prèviament descomprimides. No inclou cap funcionalitat de compressió. Per tant, la seva utilitat es limita a escenaris on les dades arriben ja comprimides en format LZSS i es necessita accelerar-ne la recuperació mitjançant la potència de la GPU.

3.3.4 CUDA-Image-and-Video-Codec

CUDA-Image-and-Video-Codec agrupa un conjunt de tècniques accelerades per GPU per a la compressió d'imatges i vídeo. El seu procés habitual consisteix a

aplicar primer una transformació espacial, com la DCT o la DWT, per analitzar les dades. A continuació, una etapa de quantització redueix la precisió de la informació menys rellevant, i finalment, una codificació d'entropia (com Huffman) empaqueta les dades de manera òptima. Tot el flux està pensat per ser paral·lelitzat a la GPU.

3.3.5 LZ4

LZ4 és un algorisme conegut per la seva excepcional velocitat. El seu mètode utilitza una taula de hash per localitzar de manera molt ràpida les seqüències de dades repetides. Aquestes seqüències són substituïdes per referències curtes, mentre que les dades no repetides es copien directament. Aquest esquema simple però eficaç li permet assolir velocitats de compressió i descompressió molt elevades, que a més poden ser accelerades amb implementacions en CUDA.

3.3.6 BZIP2.

A diferència de LZ4, BZIP2 prioritza la màxima eficiència de compressió per sobre de la velocitat. Per aconseguir-ho, aplica una seqüència de tres passos: la transformada de Burrows-Wheeler (BWT) per agrupar símbols similars, seguida de la codificació Run-Length (RLE) per a les repeticions, i finalment la codificació d'Huffman. Encara que és un mètode clàssic de CPU, la seva etapa més intensa, la BWT, s'ha intentat accelerar en algunes implementacions amb CUDA.

3.3.7 JPEG2000

JPEG2000 és una tècnica de compressió d'imatges avançada que ofereix una gran flexibilitat i qualitat. Utilitza la transformada wavelet discreta (DWT) per descompondre la imatge en diferents nivells de detall, la qual cosa permet una gestió molt eficient de la informació. Aquesta estructura, combinada amb la quantització i la codificació aritmètica, fa que sigui molt eficaç tant en mode de compressió amb pèrdua com sense pèrdua. Els seus processos més costosos poden ser accelerats considerablement amb CUDA.

4 ENTORN EXPERIMENTAL I DADES

Aquest capítol descriu els components clau sobre els quals s'ha realitzat l'estudi. S'especifica l'entorn de proves, es detallen les característiques del conjunt de dades proporcionat pel Sincrotró ALBA i s'analitza la seva entropia per establir una base teòrica sobre el seu potencial de compressió.

4.1 Entorn de proves

Les proves de compressió i descompressió es van dur a terme en un equip amb les següents especificacions de maquinari, les quals van ser considerades a l'hora d'analitzar les dificultats i el rendiment dels algorismes basats en GPU:

- Processador (CPU): Intel i5-7400
- Memòria RAM: 8 GB
- Targeta Gràfica (GPU): NVIDIA GeForce GTX 1050

4.2 Descripció de les Dades

Les dades utilitzades en aquest projecte són imatges científiques reals provinents del Sincrotró ALBA, adquirides durant experiments. Aquestes imatges tenen les següents característiques:

- Format: S'emmagatzemen en fitxers amb extensió .raw, que contenen dades pures sense cap tipus de compressió ni capçalera.
- Dimensions: Cada imatge té una dimensió constant de 4371×4150 píxels.
- Mostres per imatge: Un total de 18.144.650 mostres.
- Profunditat de bit: Cada mostra ocupa 2 bytes (16 bits) i està en format unsigned integer de 16 bits (u16be).
- Mida per fitxer: Aproximadament 36.289.300 bytes (≈35.430 KB) per imatge.
- Volum total del conjunt de dades: Es treballa amb un conjunt de 1800 imatges, la qual cosa suposa un volum total de dades d'entrada de ≈65,3 GB.

És important destacar que no són imatges convencionals; es tracta de dades científiques, sorolloses i amb poca estructura visual perceptible, però amb variacions significatives a nivell de valors de píxels. Aquesta naturalesa específica fa que els resultats de la compressió no segueixin necessàriament els patrons típics de les imatges naturals

4.3 Anàlisi de l'Entropia de les Dades

4.3.1 Entropia d'ordre zero

L'entropia, en el context de la teoria de la informació de Claude Shannon, és una mesura quantitativa de la incertesa o informació mitjana continguda en una font de dades. Indica quanta informació nova aporta cada símbol (o mostra), de manera que com més imprevisible és una mostra, més alta és la seva entropia.

Per a aquest estudi, es va calcular l'entropia d'ordre zero, que mesura la informació mitjana de cada valor sense tenir en compte l'ordre o la correlació entre mostres consecutives; s'assumeix que cada valor és independent dels altres. Per a això, es va crear un script que llegeix cada fitxer .raw, interpreta les dades com a nombres enters de 16 bits, compta la freqüència de cada valor possible (de 0 a 65535) i calcula les seves probabilitats relatives. Amb aquestes probabilitats, s'aplica la fórmula de l'entropia. Un resultat d'entropia baix indica que el fitxer conté molta redundància i, per tant, és fàcilment comprimible.

4.3.2 Resultats de l'entropia

El càlcul realitzat sobre el conjunt de dades va donar una entropia binària d'ordre zero mitjana de 0,4999 bits per mostra. Aquesta xifra és extremadament baixa si es considera que cada mostra original ocupa 16 bits. Aquest valor indica una alta redundància a les dades, la qual cosa significa que gran part de la informació és repetitiva o previsible. En conseqüència, es pot preveure que aquestes dades són molt favorables a la compressió i que els algorismes haurien de ser capaços de reduir significativament la mida dels fitxers originals. Aquest valor serveix com a límit inferior teòric per a qualsevol algorisme de compressió sense pèrdua.

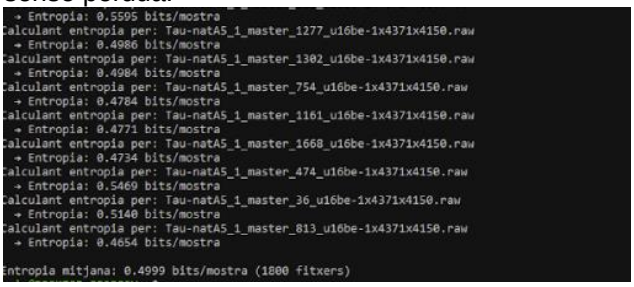

Fig.1: Entropia mitjana

5 ENTORN EXPERIMENTAL I DADES

Aquest capítol detalla el procés d'execució de les proves i presenta una anàlisi exhaustiva dels resultats obtinguts. Primer es descriu la metodologia seguida per a l'avaluació dels algorismes i, a continuació, s'analitzen les mètriques de rendiment clau.

5.1 Metodologia Experimental

Per a l'avaluació de cada algorisme, es va seguir un procés sistemàtic per garantir la consistència i fiabilitat de les dades recollides. En una fase prèvia, es van crear scripts específics per a cada algorisme amb l'objectiu de calcular automàticament el temps de compressió i descompressió, així com per emmagatzemar els fitxers resultants en carpetes separades per a una anàlisi posterior.

Per garantir la fiabilitat de les mesures, cada prova es va executar múltiples vegades. Els resultats presentats corresponen a la mitjana obtinguda, permetent una avaluació robusta de l'eficàcia dels algorismes. Durant les proves, també es va monitorar l'ús de recursos computacionals per detectar possibles colls d'ampolla, tant en les execucions sobre CPU com sobre GPU. Aquesta metodologia va proporcionar una base sòlida per a la comparació de rendiment entre les diferents tècniques.

5.2 Anàlisi de resultats

A continuació, es presenten i s'analitzen els resultats obtinguts per a cada mètrica de rendiment avaluada.

5.2.1 Temps de compressió i descompressió

Aquesta mètrica mesura el temps total, en minuts, que cada algorisme va requerir per comprimir i descomprimir el conjunt complet de 1800 imatges. Es van realitzar dues execucions completes per garantir la consistència dels resultats i la fiabilitat de les mesures. Aquesta repetició permet assegurar que les conclusions extretes són objectives i significatives.
Execució 1:

Algorisme	Temps de compressió(min)	Temps de descompressió (min)	Total(min)
CUDA_BSC	45,93	51,39	97,32
CUDA-ICPP	58,95	17,26	76,21
CUDA-LZSS-UNKNOWN	-	157,53	-
CUDA-IMAGE-and-video-codec	11,14	44,63	55,77
LZ4	239,45	14,55	254,00
BZIP2	59,90	34,25	94,15
JPEG2000	100,20	46,80	147,00

Taula 1: Temps primera execució

Execució 2:

Algorisme	Temps de compressió(min)	Temps de descompressió (min)	Total(min)
CUDA_BSC	46,80	52,10	98,90
CUDA-ICPP	59,74	18,36	78,20
CUDA-LZSS-UNKNOWN	-	160,47	-
CUDA-IMAGE-and-video-codec	11,59	43,62	55,21
LZ4	237,33	15,65	252,98
BZIP2	57,69	35,42	93,11
JPEG2000	100,45	45,73	146,18

Taula 2: Temps segona execució

Respecte als temps d'execució, CUDA_BSC mostra un temps de compressió que es manté relativament estable entre les dues execucions, amb 45,93 minuts a la primera execució i 46,80 minuts a la segona, el que implica que la compressió és consistent. Els temps de descompressió també són lleugerament més elevats en la segona execució (51,39 minuts a la primera i 52,10 minuts a la segona), però el total de temps és bastant similar, amb una variació de només 1,58 minuts entre les dues execucions. Això suggereix que CUDA_BSC ofereix una bona consistència en els seus temps de compressió i descompressió, amb una mitjana d'uns 97 minuts per a la compressió i descompressió combinades.

CUDA-ICPP té un temps de compressió que és bastant més llarg que el de CUDA_BSC, amb 58,95 minuts a la primera execució i 59,74 minuts a la segona. El temps de descompressió és relativament curt, amb 17,26 minuts a la primera execució i 18,36 minuts a la segona, indicant que la descompressió d'aquest algorisme és força ràpida. El total de temps és de 76,21 minuts a la primera execució i 78,20 minuts a la segona, amb una petita variació, destacant la rapidesa de descompressió en comparació amb la compressió.

CUDA-LZSS-UNKNOWN no proporciona dades sobre el temps de compressió, però el temps de descompressió és notablement llarg (157,53 minuts a la primera execució i 160,47 minuts a la segona), la qual cosa suggereix que aquest algorisme pot ser més lent en termes de descompressió. La manca de dades sobre la compressió fa difícil avaluar el rendiment global de l'algorisme.

CUDA-IMAGE-and-video-codec és un dels algorismes més ràpids en compressió, amb només 11,14 minuts en la primera execució i 11,59 minuts en la segona. El temps de descompressió és relativament llarg (44,63 minuts a la primera execució i 43,62 minuts a la segona), tot i que encara es manté per sota del temps de compressió. El total de temps és força consistent entre les dues execucions, amb 55,77 minuts a la primera i 55,21 minuts a la segona, demostrant que aquest algorisme és molt eficient en la compressió, tot i que la descompressió requereix més temps.

LZ4 es destaca per la seva velocitat en descompressió, amb temps de només 14,55 minuts en la primera execució i 15,65 minuts en la segona. No obstant això, el temps de compressió és extremadament llarg (239,45 minuts a la primera execució i 237,33 minuts a la segona), el que indica que LZ4 pot no ser ideal en escenaris on es necessiti una compressió ràpida. El total de temps per LZ4 és significativament més alt que la resta, amb 254,00 minuts a la primera execució i 252,98 minuts a la segona.

BZIP2 presenta un temps de compressió relativament llarg (59,90 minuts a la primera execució i 57,69 minuts a la segona), però els seus temps de descompressió són molt més ràpids (34,25 minuts a la primera execució i 35,42 minuts a la segona). El total de temps per BZIP2 és de 94,15 minuts a la primera execució i 93,11 minuts a la segona, indicant una eficiència moderada en termes de temps de compressió/descompressió combinada.

JPEG2000 és un dels algorismes més lents, amb temps de compressió de 100,20 minuts a la primera execució i 100,45 minuts a la segona. A pesar que els temps de descompressió (46,80 minuts a la primera execució i 45,73 minuts a la segona) són una mica més ràpids, el total de temps (147,00 minuts a la primera execució i 146,18 minuts a la segona) és el més alt de tots. Això indica que, tot i que JPEG2000 pot ser útil en compressió d'imatges d'alta qualitat, no és el més eficient en termes de temps total.

5.2.2 Mida dels fitxers comprimits

Aquesta mètrica compara la mida total dels fitxers després de la compressió respecte a la mida original descomprimida (35.430 KB per imatge).

Algorisme	Tamany arxiu comprimit (KB)	Tamany arxiu descomprimit (KB)
CUDA_BSC	1.107	35.430
CUDA-ICPP	4.669	35.430
CUDA-LZSS-UNKNOWN	-	35.430
CUDA-IMAGE-and-video-codec	12.200	35.430
LZ4	2.763	35.430
BZIP2	17.500	35.430
JPEG2000	1.862	35.430

Taula 3: Mida dels fitxers

L'anàlisi de les mides dels fitxers comprimits revela una considerable variabilitat en l'eficiència de cada algorisme per reduir l'espai d'emmagatzematge. En aquest aspecte, CUDA_BSC es posiciona com l'algorisme més eficaç, aconseguint la mida comprimida més petita de totes amb només 1.107 KB. El seguei-xen a una certa distància JPEG2000, amb 1.862 KB, i LZ4, que produeix arxius de 2.763 KB, tots dos oferint una compressió notable.

En una posició intermèdia es troba CUDA-ICPP, que presenta una mida comprimida de 4.669 KB, el que demostra una reducció decent de gairebé el 87% respecte a l'original. A l'altre extrem, altres algorismes prioritzen diferents aspectes per sobre de la mida final. CUDA-IMAGE-and-video-codec genera arxius considerablement més grans (12.200 KB), la qual cosa suggereix un enfocament en la velocitat o la qualitat per sobre d'una compressió agressiva. BZIP2, per la seva banda, produeix els fitxers de major mida (17.500 KB), un resultat que s'entén com el cost associat al seu mètode de compressió, conegut per ser eficaç però exigent en temps.

Finalment, no va ser possible realitzar una comparació directa per a CUDA-LZSS-UNKNOWN, ja que no es van poder obtenir dades de la seva mida comprimida.

5.2.3 Ratió de compressió

El ratio de compressió és una mesura que descriu l'efectivitat d'un algorisme de compressió a l'hora de reduir la mida d'un arxiu. Es calcula dividint la mida original de l'arxiu entre la mida comprimida. Com més gran sigui la reducció de la mida de l'arxiu, millor serà la compressió.

Algorisme	Ratio de compressió
CUDA_BSC	3,12%
CUDA-ICPP	13,17%
CUDA-LZSS-UNKNOWN	-
CUDA-IMAGE-and-video-codec	34,43%
LZ4	7,79%
BZIP2	49,39%
JPEG2000	5,25

Taula 4: Ratió de compressió

En aquest anàlisi de ratios de compressió, es pot veure que els algorismes mostren resultats força variats pel que fa a la seva capacitat per reduir la mida dels arxius originals.

Sobresurt BZIP2 com l'opció més eficient en termes de compressió pura, aconseguint el millor resultat de tots amb un ratio del 45,49%. Això significa que redueix la mida de l'arxiu a gairebé la meitat, tot i que aquesta gran eficiència té com a contrapartida un temps de processament més lent. El segueix CUDA-IMAGE-and-video-codec, que ofereix una compressió substancial i molt atractiva del 34,43%, sent una opció excel·lent quan es necessita una reducció important de mida amb un temps d'execució acceptable.

En una posició intermèdia es troba CUDA-ICPP, amb un ratio del 13,17%, el que indica un rendiment moderat i útil per a situacions que busquen un equilibri entre compressió i temps d'execució. Per la seva banda, LZ4 no destaca per la seva compressió (7,79%), però aquest valor és secundari al seu principal avantatge, que és la velocitat de descompressió, fent-lo ideal per a l'accés ràpid a les dades.

Finalment, a la part baixa de l'eficiència per a aquest conjunt de dades es troben JPEG2000, amb un 5,25%, i especialment CUDA_BSC, amb només un 3,12%. Aquests resultats demostren que, malgrat altres qualitats tècniques, no són les opcions adequades si l'objectiu principal és aconseguir una gran reducció de la mida dels fitxers.

Aquest anàlisi revela que, mentre alguns algorismes sobresurten en la compressió pura, altres ofereixen avantatges en velocitat o en la capacitat per manejar tipus de dades específics. Cada algorisme té els seus punts forts i febles, depenent de les prioritats de l'usuari.

5.2.4 Bits per sample

El concepte de bits per sample (bps) indica quants bits es necessiten per representar una unitat de dada (o mostreig) en el conjunt de dades després de la compressió. És una mesura de l'eficàcia de la compressió en termes de l'espai necessari per representar cada mostreig de les dades. Aquesta mesura permet comparar els algorismes de compressió en termes de la quantitat d'informació que es conserva per cada mostreig un cop aplicat l'algorisme. Com menys bits per mostra, millor serà la compressió.

Algorisme	Bits per sample (bps)
CUDA_BSC	0,45 bps
CUDA-ICPP	1,89 bps
CUDA-LZSS-UNKNOWN	1,68 bps
CUDA-IMAGE-and-video-codec	4,54 bps
LZ4	1,03 bps
BZIP2	6,52 bps
JPEG2000	0,69 bps

Taula 5: Bits per sample

A la Taula 5 es presenten els valors de bits per mostra obtinguts per a cada algorisme.
L'anàlisi dels resultats de la taula mostra una clara jerarquia en l'eficiència dels algorismes. De manera destacada, CUDA_BSC es posiciona com el més eficient amb un valor excepcional de 0,45 bps, demostrant una compressió molt potent a nivell de mostra. El segueix JPEG2000, amb un resultat també molt competitiu de 0,69 bps, que reforça la seva qualitat per a dades visuals. En una posició intermèdia, LZ4 (1,03 bps) ofereix un bon equilibri, mentre que CUDA-LZSS-UNKNOWN (1,68 bps) i CUDA-ICPP (1,89 bps) presenten una eficiència raonable, encara que menor. Finalment, CUDA-IMAGE-and-video-codec (4,54 bps) i, especialment, BZIP2 (6,52 bps) són els menys eficients en aquesta mètrica. Aquests resultats evidencien que una alta compressió global, com la que se sol associar a BZIP2, no sempre garanteix una representació compacta per a cada mostra individual.

5.2.5 Mega samples per segon (MSamples/s)

La mesura de Mega Samples per segon és un indicador de rendiment dels algorismes de compressió, mostrant la quantitat de mostres comprimides per segon. Aquest valor es calcula a partir del temps de compressió total, tenint en compte el nombre de mostres que s'han de comprimir.

A continuació es mostra el procés de càlcul i la comparació entre els diferents algorismes:
Dades conegudes:

- Mostres per imatge = 4371 × 4150 = 18.144.650
- Núm. d'imatges = 1800
- Total de mostres = 18.144.650 × 1800 = 32.660.370.000

Algorisme	Temps de compressió(min)	Temps de compressió(s)	Msamples/s
CUDA_BSC	46,37	2782,2	11,74
CUDA-ICPP	59,34	3560,4	9,17
CUDA-IMAGE-and-video-codec	11,36	681,6	47,91
LZ4	238,39	14303,4	2,28
BZIP2	58,50	3528,0	9,26
JPEG2000	100,33	6019,8	5,43

Taula 6: MSample/s

D'entre tots els algorismes analitzats, CUDA-IMAGE-and-video-codec destaca clarament com el més ràpid, amb una velocitat de compressió de 47,91 Msamples/s. Aquesta diferència és molt significativa, ja que comprimeix gairebé quatre vegades més ràpid que el següent algorisme, i això el fa especialment adequat per a entorns on el temps de processament és crític, com ara el tractament de dades massives o en temps real. A força distància se situa CUDA_BSC, amb 11,74 Msamples/s, que tot i no igualar el rendiment de CUDA-IMAGE-and-video-codec, ofereix una velocitat força elevada i coherent amb el fet de ser una implementació GPU, mostrant un bon balanç entre rendiment i simplicitat d'ús.

Molt a prop entre si trobem BZIP2 i CUDA-ICPP, amb valors pràcticament iguals de 9,26 i 9,17 Msamples/s respectivament. Tot i que un és una solució tradicional basada en CPU i l'altre es beneficia de l'acceleració GPU, la proximitat dels resultats mostra que el rendiment també pot dependre fortament de com està optimitzat l'algorisme. En aquest cas, ambdós ofereixen una compressió raonablement ràpida i adequada per a aplicacions que requereixen un bon equilibri entre velocitat i eficiència de compressió.

Pel que fa a JPEG2000, amb 5,43 Msamples/s, se situa clarament per sota dels anteriors. Tot i això, aquest algorisme no està pensat exclusivament per al rendiment, sinó que destaca per la qualitat i eficiència de compressió, especialment en imatges. És una opció preferent quan la precisió de la compressió, la qualitat del resultat final i la gestió visual són prioritàries respecte a la velocitat.

Finalment, LZ4, amb tan sols 2,28 Msamples/s, és el més lent en compressió dels algorismes analitzats. Aquesta dada confirma que, malgrat la seva fama per la rapidesa en descompressió, no és la millor opció quan es busca velocitat en el procés de compressió inicial. No obstant, pot seguir sent útil en escenaris on la compressió es fa una única vegada i les dades es consulten moltes.

5.3 Dificultats trobades

Durant el desenvolupament de la fase experimental, es van presentar diverses dificultats tècniques relacionades principalment amb la instal·lació, configuració i execució dels algorismes basats en GPU. A continuació, es descriuen els principals problemes trobats i els intents de solució.

Problemes de stack amb cuda-lzss-cluster: aquest algorisme va presentar errors greus relacionats amb l'ús de la pila de memòria (stack), els quals provocaven fallides una fallida a l'hora de comprimir. Malgrat haver intentat diverses solucions, com modificar la mida de la pila o ajustar la configuració del compilador, no s'ha pogut resoldre aquest problema de manera satisfactòria.

Versió desactualitzada de cuda-bzip2: el repositori oficial d'aquest algorisme presenta una versió obsoleta que genera errors durant la compilació i execució, relacionats amb incompatibilitats amb versions recents de CUDA i del compilador. Davant d'això, es va intentar utilitzar una versió alternativa disponible a GitHub (cuda_bzip2 del repositori d'aditya12agd5), però aquesta tampoc va poder ser executada correctament, ja que no es tractava d'una versió oficial i que mancava de dependències difícils de resoldre.

Compatibilitat amb CUDA i entorn de desenvolupament: Diversos dels algorismes GPU provats van presentar incompatibilitats amb la versió de CUDA instal·lada, la qual cosa va generar problemes de dependències. Finalment vaig poder resoldre els problemes borrant totes les dependències de CUDA que tènia i tornant a instalar-ho amb la versió de l'algorisme recomanada.

Degut a aquests problemes, s'ha experimentat amb més algorismes per veure si era possible executar-ne algun més. Es van provar GPUHD, nvCOMP i dietgpu.

En el cas de GPUHD, l'algorisme trigava més de 6 hores en comprimir i descomprimir els fitxers, cosa que el feia poc viable per al nostre cas d'ús. Aquest temps d'execució tan elevat es devia, en part, a un coll d'ampolla a la CPU, l'ús de la qual arribava al 100% durant el procés.

Pel que fa a nvCOMP, es va intentar instal·lar el paquet mitjançant pip, però es va produir un problema de dependències: certes biblioteques requerides, com nvidia-nvcomp-cu12, no estaven disponibles per a la versió del sistema ni del Python instal·lat, i no hi havia cap paquet precompilat compatible.

Finalment, s'ha provat dietgpu, l'execució del comandament docker build -t dietgpu . donava un error indicant que el contenidor estava deprecat tal i com es veu a la Figura 2 i que no era compatible amb les versions recents del Docker o de CUDA, fet que impossibilitava la seva instal·lació i ús.

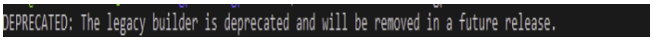

Fig.2: Algorisme depcrecated

6 CONSIDERACIÓ DE LA VIABILITAT DE CADA ALGORISME EN EL CONTEXT DEL SINCOTRÓ D'ALBA

En aquest capítol s'analitza el rendiment de cada algorisme en el context específic del Sincrotró ALBA, considerant les seves fortaleses i debilitats a partir de les mètriques obtingudes. Aquesta anàlisi de viabilitat culmina amb una sèrie de recomanacions pràctiques per a la selecció de l'eina més adequada segons diferents escenaris d'ús, i es tanca amb una anàlisi visual global del rendiment.

6.1. Anàlisi detallada de viabilitat per algorisme

L'anàlisi de viabilitat per al context del Sincrotró ALBA revela perfils de rendiment molt diversos, on cada algorisme destaca en un nínxol particular. Entre totes les opcions, CUDA-IMAGE-and-video-codec es presenta com la solució més viable i equilibrada per a un ús general. El seu rendiment en la compressió és excepcionalment ràpid, amb un temps d'aproximadament 11 minuts. Aquesta velocitat es complementa amb un bon ratio de compressió (34,43%) i un temps de descompressió acceptable (uns 44 minuts). Aquesta combinació de rapidesa i eficiència el fa ideal per a situacions on es necessiten resultats àgils sense sacrificar un estalvi considerable d'espai.

Per a necessitats més específiques, altres algorismes es defineixen com a eines especialitzades. LZ4, per exemple, és la solució idònia quan la prioritat absoluta és la descompressió ultra ràpida (uns 15 minuts), tot i que el seu ús es veu limitat per un temps de compressió extremadament lent, superior a les 4 hores. En el pol oposat.

Un cas particularment interessant és el de CUDA_BSC. Tot i que els seus temps de procés són llargs i el seu ratio de compressió és baix, la seva viabilitat no es mesura en velocitat, sinó en la seva superioritat tècnica. És l'únic algorisme que aconsegueix una taxa de bits per mostra (0,45 bps) inferior a l'entropia teòrica de les dades (0,4999 bps). Aquesta fita garanteix la màxima fidelitat a la informació, convertint-lo en una eina molt valuosa per a anàlisis científiques sensibles on la puresa de les dades és la prioritat absoluta.

Finalment, altres algorismes completen el panorama. CUDA-ICPP es posiciona com una solució intermèdia viable, especialment si es busca una descompressió ràpida (18 minuts) amb una millor compressió que LZ4. En canvi, JPEG2000 i CUDA-LZSS-UNKNOWN es consideren opcions poc viables per a l'entorn d'ALBA. El primer, pels seus llargs temps de processament i baix ratio, i el segon, pel seu temps de descompressió desmesurat (més de 157 minuts) i la manca de dades completes.

6.2. Recomanacions per escenari d'ús

A partir de l'anàlisi anterior, es poden establir les següents recomanacions pràctiques:

- Per a un equilibri entre Velocitat i Reducció de Mida: CUDA-IMAGE-and-video-codec és la millor opció per la seva compressió ràpida i bon ratio.
- Si la prioritat és la Descompressió Ràpida: LZ4 és l'opció indiscutible per accedir a les dades de manera gairebé instantània.
- Si la prioritat és la Màxima Reducció de Mida: CUDA_BSC és l'algorisme a escollir, ja que ofereix el millor ratio de compressió (3,12%), ideal per a emmagatzematge a llarg termini.
- Per a Escenaris Experimentals o de Màxima Fidelitat: CUDA_BSC torna a ser la millor elecció per la seva capacitat única de comprimir per sota de l'entropia, garantint la màxima puresa de la informació.

6.3. Anàlisi gràfica del rendiment global

Tot i que una anàlisi superficial podria suggerir un equilibri entre diferents factors com la velocitat i el ratio, una avaluació més profunda dels resultats revela que CUDA_BSC es posiciona com l'algorisme tècnicament més avançat i eficient per a aquest conjunt de dades específic. La raó principal rau en la seva característica única i excepcional: és l'únic algorisme dels estudiats que aconsegueix una xifra de bits per mostra (0,45 bps) inferior a l'entropia d'ordre zero teòrica de les dades (0,4999 bits/mostra).

Aquest fet és particularment rellevant, ja que l'entropia d'ordre zero estableix el límit teòric de compressió per a qualsevol algorisme que tracti les mostres com a valors independents. En superar aquest límit, CUDA_BSC demostra que no només explota la freqüència estadística dels valors, sinó que també és capaç de capturar i modelar correlacions d'ordre superior presents a les dades. Aquesta eficiència

s'atribueix directament a la seva metodologia, que combina la Transformada de Burrows-Wheeler (BWT) per reordenar el flux de dades i agrupar patrons, seguida d'altres etapes de codificació.

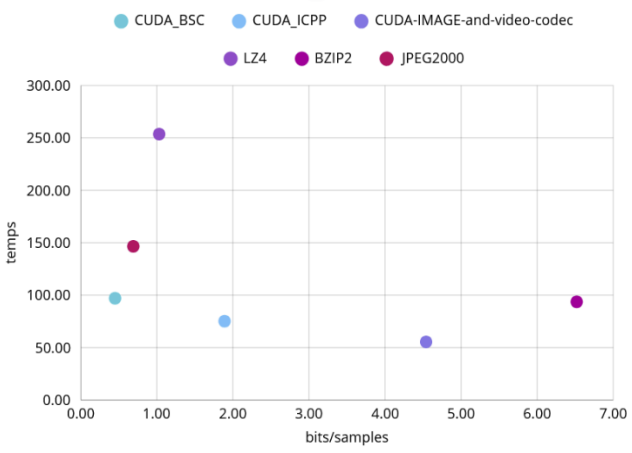

Fig.3: Comparativa de rendiment dels algorismes.

L'eix X representa l'eficiència de la compressió (bits per mostra) i l'eix Y representa el temps total de processament (compressió + descompressió) en minuts. Un punt ideal se situaria a la cantonada inferior esquerra (mínim temps, màxima compressió).

Com s'il·lustra a la Figura 3, on es representa el temps de processament en funció de l'eficiència de compressió, l'algorisme CUDA_BSC destaca de manera significativa. Se situa en la posició més a l'esquerra del gràfic, la qual cosa indica que assoleix la representació de dades més compacta de totes les opcions avaluades, amb el valor de bits per mostra més baix (0,45 bps).

Si bé el seu temps d'execució (eix Y) no és el més ràpid en termes absoluts, la seva capacitat per assolir una representació tan eficient i fidel de les dades originals el converteix en la solució tècnicament més notable. Aquesta qualitat és especialment valuosa en entorns científics on garantir la màxima fidelitat i eliminar qualsevol redundància artificial és una prioritat per sobre de la velocitat pura. Per tant, la seva posició a la gràfica confirma que, des del punt de vista de la teoria de la informació, CUDA_BSC ofereix una qualitat i eficiència de compressió superior a la resta.

7 Metodologia

La metodologia seguida en aquest TFG es va estructurar en quatre fases consecutives per garantir un procés rigorós des de la recerca inicial fins a la redacció final. El desenvolupament del projecte va combinar treball teòric i experimental.

Fase 1: Investigació inicial dels algorismes de compressió. En la primera fase es va dur a terme una recerca bibliogràfica i tècnica per identificar algorismes de compressió amb potencial aplicació. Es van considerar l'eficiència, la compatibilitat amb CUDA i l'adequació a grans volums de dades. Aquesta etapa va culminar amb la selecció d'un conjunt d'algorismes candidats.

Fase 2: Execució i avaluació dels algorismes. La segona fase va tenir un caràcter experimental. Es van implementar i configurar els algorismes seleccionats sobre conjunts de dades reals del Sincrotró ALBA. Es van executar sessions de compressió i descompressió, repetides per garantir la consistència, i es van recollir mètriques detallades de rendiment.

Fase 3: Anàlisi de resultats i conclusions.La tercera fase es va centrar en l'anàlisi crítica dels resultats obtinguts. Es van comparar detalladament els diferents algorismes, tenint en compte dades quantitatives i requisits pràctics de l'ALBA. S'han identificat les fortaleses i debilitats de cada enfocament per establir quins algorismes són realment viables.

Fase 4: Redacció de la memòria del TFG. Finalment, la quarta fase va correspondre a la redacció formal d'aquesta memòria. Va incloure la descripció completa de les fases anteriors, l'explicació detallada dels algorismes, els resultats obtinguts i les conclusions finals, incorporant taules i gràfics per facilitar-ne la comprensió.

7.1 Diagrama de Gantt

Per tal d'organitzar el desenvolupament del projecte de manera estructurada i eficient, s'ha fet servir un diagrama de Gantt que reflecteix les diferents fases de treball, les tasques associades a cadascuna i l'estat d'execució corresponent. Aquest diagrama ha servit com a eina de planificació i seguiment, permetent establir prioritats, distribuir el temps de manera equilibrada i assegurar l'assoliment progressiu dels objectius del treball. La planificació es divideix en quatre fases principals: la investigació inicial, l'execució i avaluació dels algorismes, l'anàlisi de resultats i la redacció de la memòria final.

8 Conclusions

Aquest treball ha permès dur a terme una anàlisi exhaustiva de diversos algorismes de compressió, tant basats en CPU com en GPU, aplicats a dades científiques del Sincrotró ALBA. A partir dels resultats obtinguts, es poden extreure les següents conclusions generals:

1. No existeix una solució universal, sinó un algorisme òptim per a cada escenari. L'anàlisi de viabilitat demostra que la selecció de l'algorisme ha d'estar alineada amb els requisits específics de cada cas d'ús. Per a escenaris que requereixen un equilibri entre velocitat de compressió i una bona reducció de mida,

CUDA-IMAGE-and-video-codec és la millor opció. Si la prioritat absoluta és la velocitat de descompressió per a un accés quasi immediat a les dades, LZ4 és l'elecció adequada, malgrat la seva baixa eficiència de compressió. Per a l'emmagatzematge a llarg termini, on la màxima reducció de mida és crucial, BZIP2 ofereix el millor ratio de compressió (45,49%), encara que els seus temps de procés són elevats.

2. CUDA_BSC és l'algorisme tècnicament més avançat i eficient. Malgrat que altres algorismes el superen en velocitat o en ratio de compressió brut, CUDA_BSC ha demostrat una superioritat tècnica excepcional. És l'únic algorisme avaluat que aconsegueix una representació de menys bits per mostra (0,45 bps) que l'entropia d'ordre zero de les dades (0,4999 bps). Això indica que és capaç de modelar correlacions complexes entre les dades, oferint la compressió més fidel a la informació real continguda als fitxers.

3. La viabilitat pràctica dels algorismes depèn de l'equilibri de totes les seves mètriques. Algorismes com CUDA-ICPP es presenten com a solucions de nínxol, viables si es busca una descompressió ràpida amb una millor compressió que LZ4. D'altres, com JPEG2000 o CUDA-LZSS-UNKNOWN, es consideren poc viables per a l'entorn d'ALBA a causa dels seus elevats temps de processament, que no compensen els seus beneficis.

4. La implementació d'algorismes de compressió en GPU presenta reptes tècnics significatius. L'estudi ha posat de manifest que l'ecosistema d'eines de compressió accelerades per GPU encara no és del tot madur. S'han trobat dificultats notables com ara repositoris obsolets (cuda-bzip2), errors de gestió de memòria no resolts (cuda-lzss-cluster), problemes de dependències amb les versions de CUDA i Python (nvCOMP) i l'ús de contenidors deprecats (dietgpu). Aquests obstacles demostren que, més enllà del rendiment teòric, la facilitat d'integració i el manteniment del programari són factors crítics a considerar.

5. La metodologia aplicada ha permès complir els objectius del treball. Malgrat les dificultats, s'ha realitzat una anàlisi comparativa robusta que ha permès avaluar el rendiment real de cada algorisme i extreure conclusions fonamentades i directament aplicables a les necessitats del Sincrotró ALBA.

9 BIBLIOGRAFÍA

NVIDIA Corporation. (2023). CUDA Programming Guide. NVIDIA. Disponible a: https://docs.nvidia.com/cuda/cuda-c-programming-guide/

GitHub - NDZip. (s.d.). NDZip - GPU accelerated lossless compression. Disponible a: https://github.com/celerity/ndzip

GitHub - CUDA Image and Video Codec. (s.d.). CUDA Image and Video Codec Library. Disponible a: https://github.com/13Karl/CUDA-Image-and-Video-codec

GitHub - GPU Lossless Compression. (s.d.). GPU Lossless Compression Project. Disponible a: https://github.com/dingwentao/GPU-lossless-compression

LZ4. (2024). LZ4 - Extremely fast compression. Disponible a: https://github.com/lz4/lz4

Bzip2. (2024). bzip2 Home Page. Disponible a: https://www.sourceware.org/bzip2/

OpenJPEG. (2024). OpenJPEG - Open-source JPEG 2000 codec. Disponible a: https://github.com/uclouvain/openjpeg

Kitware, Inc. (2024). CMake Installation Guide. Disponible a: https://cmake.org/install/

NVIDIA Corporation. (2024). CUDA Installation Guide for Linux. Disponible a: https://docs.nvidia.com/cuda/cuda-installation-guide-linux/

NVIDIA Corporation. (2024). CUDA Installation Guide for Microsoft Windows. Disponible a: https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/

Salomon, D. (2007). Data Compression: The Complete Reference. Springer.

NVIDIA Developer Forums. (s.d.). CUDA streams and error handling: Discussió sobre la gestió d'errors en l'ús de fluxos (streams) en CUDA, amb informació sobre errors "sticky" i com manejar-los. Disponible a: https://forums.developer.nvidia.com/t/cuda-streams-and-error-handling/276081

aditya12agd5. (s.d.). cuda_bzip2 - CUDA implementation of bzip2 compression algorithm. GitHub. Disponible a: https://github.com/aditya12agd5/cuda_bzip2

GitHub - GPUHD. (s.d.). GPUHD - High Definition GPU Compression. Disponible a: https://github.com/weissenberger/gpuhd

NVIDIA. (s.d.). nvCOMP - GPU accelerated compression library. Disponible a: https://developer.nvidia.com/nvcomp i https://github.com/NVIDIA/nvcomp

Facebook Research. (s.d.). DietGPU - GPU-accelerated lossless compression. Disponible a: https://github.com/facebookresearch/dietgpu

Juan Enredado. (2021). La entropía de Shannon como medida de la incertidumbre y la información potencial (Parte II). Medium. Disponible a: https://medium.com/@JuanEnredado/la-entrop%C3%ADa-de-shannon-como-medida-de-la-incertidumbre-y-la-informaci%C3%B3n-potencial-parte-ii-fc5e32a2b80e

Speccy Wiki. (s.d.). Compresión RLE: Ratio de compresión. Disponible a: https://wiki.speccy.org/cursos/ensamblador/compresion_rle

TutorialsPoint. (s.d.). Concept of Bits per Pixel. Disponible a: https://www.tutorialspoint.com/dip/concept_of_bits_per_pixel.htm

Refraction Productions. (s.d.). ¿Qué es frecuencia de muestreo, profundidad o resolución en bits de audio? Disponible a: https://refractionproductions.com/que-es-frecuencia-de-muestreo-profundidad-resolucion-bits-audio/

APÈNDIX

Nom de la tasca	Estat	Setmana 1	Setmana 2	Setmana 3	Setmana 4	Setmana 5	Setmana 6	Setmana 7	Setmana 8	Setmana 9	Setmana 10	Setmana 11	Setmana 12	Setmana 13	Setmana 14	Setmana 15	Setmana 16	Setmana 17	Setmana 18	Setmana 19	Setmana 20	Setmana 21
Fase 1: Investigació inicial dels algoritmes de compressió	Acabat																					
Recerca de possibles algoritmes de compressió	Acabat																					
Estudi de la compatibilitat dels algoritmes amb la infraestructura	Acabat																					
Selecció d'algoritmes	Acabat																					
Documentació de les característiques bàsiques dels algoritmes seleccionats	Acabat																					
Fase 2: Execució i avaluació dels algoritmes	Acabat																					
Implementació i configuració dels algoritmes seleccionats	Acabat																					
Preparació dels conjunts de dades (fitxers del Sincrotró ALBA)	Acabat																					
Execució de les proves de compressió/descompressió per a cada algorisme	Acabat																					
Repetició de les proves per garantir la fiabilitat dels resultats	Acabat																					
Anàlisi preliminar dels resultats obtinguts	Acabat																					
Creació d'una taula comparativa dels resultats (temps de compressió, mida comprimida, etc.)	Acabat																					
Fase 3: Anàlisi de resultats i conclusions	Acabat																					
Anàlisi detallada dels resultats obtinguts en la fase 2	Acabat																					
Comparació del rendiment entre els diferents algoritmes	Acabat																					
Identificació dels punts forts i febles de cada algorisme	Acabat																					
Consideració de la viabilitat de cada algorisme en el context del Sincrotró ALBA	Acabat																					
Elaboració de recomanacions per a la selecció d'algoritmes en funció dels requisits	Acabat																					
Fase 4: Redacció de la memòria del TFG	Acabat																					
Redacció del contingut de la memòria (introducció, metodologia, etc.)	Acabat																					
Descripció de les proves realitzades	Acabat																					
Descripció detallada dels resultats i la seva anàlisi	Acabat																					
Creació de gràfics comparatius i taules resum	Acabat																					
Preparació del material per a la presentació (gràfics, taules, visualitzacions)	Acabat																					
Redacció de les conclusions finals	Acabat																					

Fig.4: Diagrama de gantt