

Commissione Scientifica Nazionale 1 - INFN

Report alla CSN1 sull'attività del progetto GAP



Gianluca Lamanna
(P.I. - INFN sez. di Pisa)

Massimiliano Fiorini
(Università di Ferrara)

Andrea Messina
(Università di Roma "Sapienza")

Indice

Introduzione	1
1 Il Progetto GAP	2
1.1 Scopi e motivi del Progetto GAP	3
1.2 Confronto con altre tecniche di processamento in tempo reale	7
1.3 Stato dell'arte delle GPU	9
1.4 Profilo economico del progetto	10
2 Unità di Ricerca e collaborazioni	12
2.1 Unità INFN di Pisa	12
2.2 Unità di Ferrara	14
2.3 Unità di Roma	15
2.4 Riunioni periodiche e riunioni nazionali	17
3 Acquisti e Personale	19
4 GPU e L0: stato	21
4.1 Ricostruzione di anelli nel contatore RICH	21
4.2 Misura e riduzione della latenza	23
4.2.1 PFRING	24
4.2.2 NaNet	26
4.2.3 Sistema "ibrido" per scopo di misura delle caratteristiche del PC	27
5 GPU e HLT: stato	29
5.1 Studi preliminari per il trigger HLT di ATLAS	30
5.2 Studi preliminari per il HLT trigger di LHCb	32
5.3 Studi preliminari per il trigger di NA62	32
6 Ricostruzione immagini mediche	34
6.1 Risonanza magnetica nucleare	34
6.2 Tomografia assiale (CT)	36
6.2.1 Sviluppo di un codice per le trasformazioni di proiezione e retro-proiezione per applicazioni tomografiche implementata su processori grafici NVIDIA.	36
6.2.2 Implementazione dell'algoritmo di FDK per la ricostruzione di dati di CBCT.	36
6.3 Tomografia a Positroni (PET)	37

7	Conferenze e Scuole	38
8	Problemi, difficoltà e proposte	40
9	Futuro e Conclusioni	42
	Bibliografia	43

Introduzione

In questo documento si descrive lo stato di avanzamento del progetto GAP (firb_lamanna) iniziato nell'aprile 2013 e che coinvolge tre unità di ricerca (INFN(Pisa), Roma e Ferrara). In particolare si richiameranno gli obiettivi descritti nella riunione tenuta a Bologna, a fine maggio, tra i referee e i tre responsabili del progetto, discutendo lo stato di avanzamento del progetto. Si mostreranno i risultati raggiunti e si proporranno punti di discussione per il prossimo futuro.

Il progetto è dedicato allo studio della possibilità di impiego di processori grafici (GPU) in applicazioni in real-time di fisica delle alte energie e ricostruzione di immagini nella diagnosi e la ricerca medica.

Lo scopo principale di GAP è studiare e trovare soluzioni ottimali per poter impiegare la capacità di calcolo esposta dalle GPU in applicazioni in tempo reale.

La prima parte di questo report riassume il progetto GAP, descrivendone i motivi e le scelte. Nel seguito invece si descrive la struttura dell'organizzazione del lavoro, i compiti delle unità e lo stato del progetto dopo 8 mesi dall'inizio (il progetto è iniziato ad Aprile 2013).

Capitolo 1

Il Progetto GAP

In questo paragrafo si presenterà il progetto per grandi linee confrontandolo con altre tecniche per il calcolo real-time in particolare per quello che riguarda i trigger di HEP.

Nel seguito si discuterà lo stato dell'arte dei processori grafici e le innovazioni più recenti, in modo da meglio identificare la direzione che il progetto stesso intraprenderà a causa di questi cambiamenti. Nell'ultima parte di questo paragrafo si discuterà l'iter di approvazione del progetto e i finanziamenti ricevuti.

Negli ultimi anni le GPU si sono imposte nel mondo dell'High Performance Computing (HPC) insieme ai cluster di macchine multiprocessore. La ragione alla base dello sviluppo di strutture di tipo multi e many cores per il calcolo intensivo è da ricercarsi, essenzialmente, nel fatto che l'avvicinamento dei limiti fisici nei processi di integrazione classica e le richieste di limitazione di consumo, che si traducono nella riduzione della frequenza di clock, hanno imposto di ripensare l'architettura dei computer. La preferenza verso le architetture parallele e a basso consumo è in qualche modo imposto da queste limitazioni. Dal punto di vista software questo cambio di paradigma sta richiedendo una revisione dei metodi di programmazione e un ripensamento degli algoritmi, a favore di una strutturazione più vettoriale degli stessi. In questo "ammodernamento" dell'hardware e del software per meglio sfruttare le possibilità offerte dal calcolo parallelo, le GPU, a partire dal 2007/2008, si sono imposte grazie alla loro struttura nativamente parallela. Inoltre un grande sforzo è stato fatto "esporre" le risorse hardware dei processori grafici alla programmazione standard. Questo ha portato alla definizione di standard, framework e linguaggi di programmazione per sfruttare le caratteristiche di tali processori.

Essenzialmente, rispetto ai processori normali (CPU), le GPU offrono una struttura parallela che può raggiungere le migliaia di cores con una strutturazione e velocità della memoria idonee a sfruttare l'esecuzione concorrente degli algoritmi su dati multipli. Tale architettura è ispirata, ovviamente, dalle richieste della computer grafica dove, banalizzando, i calcoli su singoli parti di un'immagini possono, in prima approssimazione, essere effettuati in modo indipendente da quelli su parti vicine. E' importante notare che lo sviluppo richiesto da questo tipo di applicazione ha stimolato, e continua a farlo, un notevole progresso nella potenza di calcolo dei singoli chip e nell'aumento della banda di memoria richiesta sulle schede grafiche e nella connessione ai PC, molto più che le richieste di intensità di calcolo scientifico.

Ciò nonostante, le ultime generazioni di schede grafiche vengono anche prodotte in

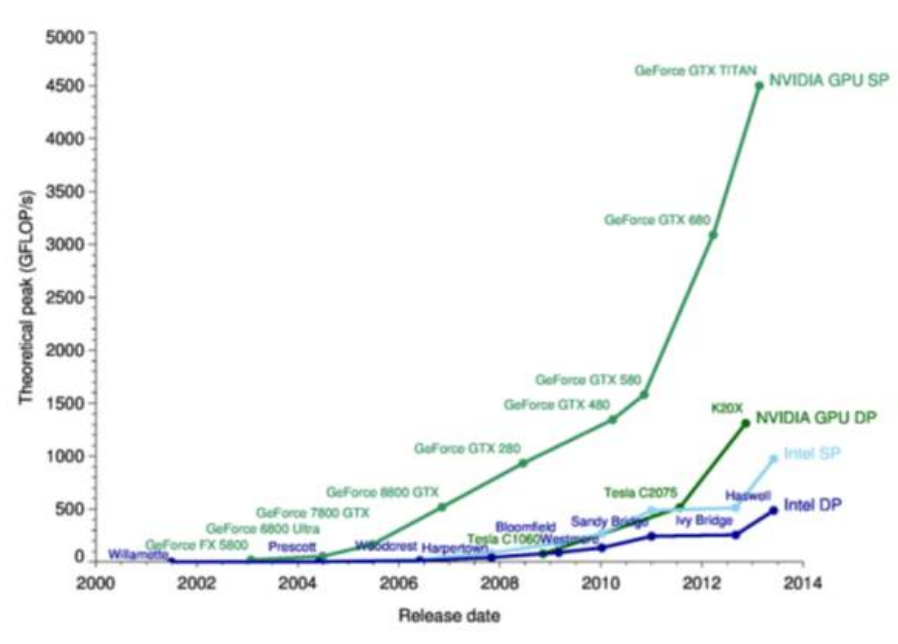


Figura 1.1: Sviluppo della capacità di calcolo delle GPU negli ultimi anni.

versioni ottimizzate per il calcolo in modo da fungere come co-processor alle operazioni di un processore tradizionale.

Il risultato di questo sviluppo è reso evidente nel grafico 1.1 che confronta la potenza di calcolo per il principale produttore di schede video (NVIDIA) e i processori Intel con architettura multicore tradizionale. Tale capacità di calcolo è, ad oggi, impiegata in moltissime applicazioni dove la parte computazionale è rilevante, dalla simulazione al test di modelli complessi, alle previsioni di diversa natura.

Nonostante l'enorme capacità di calcolo disponibile in uno spazio estremamente ridotto a costi energetici inferiori rispetto al computing classico, le caratteristiche di processamento in tempo reale dei processori grafici devono essere attentamente studiate. Il processamento delle immagini, per il quale i processori grafici sono inizialmente pensati, ammette latenze relativamente grandi, considerando che i tempi di risposti dell'occhio umano si attestano nell'ordine dei millisecondi. Applicazioni di processamento online, come quelle nel caso dei trigger di esperimenti di fisica delle alte energie, necessitano di latenze di calcolo normalmente inferiori. D'altra parte in una struttura parallela nel calcolo e nel trasporto dei dati, il problema della latenza può essere virtualmente "nascosto" sfruttando le caratteristiche concorrenziali dell'architettura stessa.

1.1 Scopi e motivi del Progetto GAP

Lo scopo di questo progetto è studiare un'architettura di selezione, analisi veloce e acquisizione dati innovativa, basata sull'utilizzo dell'informazione digitalizzata facendo uso di sistemi commerciali opportunamente adattati.

Negli esperimenti di Fisica delle Alte Energie (HEP) questo permetterebbe di applicare direttamente nei primi livelli di trigger criteri di selezione complessi, usualmente, con alcune eccezioni, adoperati soltanto nei livelli più alti. Nella ricostruzione di immagini

in contesti medicali si avrebbero tempi di elaborazione della risposta molto più brevi di quelli attuali, con grande vantaggio per la diagnostica. Gli obiettivi finali del progetto sono di dimostrare l'efficacia dello sviluppo proposto in ambiti differenti per frequenza di acquisizione dei dati, volume e intensità di calcolo: selezione in trigger sincroni di basso livello in esperimenti HEP, decisioni complesse in trigger di alto livello in HEP e ricostruzione di immagini nella diagnostica medica in Risonanza Magnetica Nucleare (NMR), Tomografia a Positroni (PET) e Tomografia Computerizzata (CT).

Un punto importante della ricerca proposta è l'utilizzo dei processori grafici (GPU), presenti nelle schede video dei computer, per calcolo in tempo reale. Tali processori sono realizzati con architettura SIMD (Single Instruction Multiple Data), particolarmente utile per l'implementazione di algoritmi paralleli. Questo, unito alla grande larghezza di banda delle schede video attuali, permette di ottenere prestazioni in termini di capacità di calcolo estremamente elevate, sfruttabili in algoritmi di pattern recognition e di ricostruzione complessa. I livelli hardware impiegati nei sistemi di trigger di HEP attuali hanno le limitazioni di essere fortemente dipendenti dagli esperimenti per i quali sono concepiti e realizzati e come tali non utilizzabili in altre applicazioni. Hanno inoltre un costo di disegno e realizzazione elevato, e mancano della capacità di elaborare le informazioni necessarie per la costruzione di primitive di qualità simile a quelle realizzabile in livelli software. Un trigger software basato sulle GPU offre risposta a tutti e tre questi problemi: da una parte si può pensare ad un'architettura indipendente dal contesto in cui è applicata, considerata l'estrema versatilità di una soluzione software, dall'altra la potenza di calcolo permette di poter prendere decisioni complesse direttamente online. Non va sottovalutato, inoltre, che l'impiego di tecnologie commerciali permette di beneficiare gratuitamente del continuo sviluppo in questo settore (editing video, video giochi, etc.), con costi di realizzazione del sistema trascurabili rispetto allo sviluppo di soluzioni ad hoc.

Le difficoltà da affrontare per poter impiegare utilmente le GPU in un contesto real-time di trigger sono legate alla necessità di un'ampia banda di trasferimento, alla velocità di calcolo necessaria per elaborare dati con frequenze delle decine MHz e, nel caso di trigger sincroni, al controllo e alla stabilità della latenza totale.

Per ottenere un trasferimento dati ad alta banda su una rete a pacchetto convenzionale (ad es. ethernet), investigheremo due possibili alternative: l'impiego delle nuove NIC (interfacce di rete) a 10 Gb/s o 40 Gb/s e la realizzazione di una scheda di rete in cui, utilizzando opportunamente delle FPGA, sia possibile effettuare operazioni di off-load del protocollo e pre-processamento dei dati.

Per ottimizzare la banda di trasferimento e limitare la variabilità della latenza dovuta al sistema operativo si svilupperà un driver ad alte prestazioni, in collaborazione con una ditta esterna con competenze nel settore.

Nei livelli sincroni di trigger la stabilità della latenza nella decisioni è fondamentale per il funzionamento del sistema. Dopo un tempo predefinito e fisso in seguito all'arrivo dell'evento nelle schede di acquisizione, la decisione di trigger deve immancabilmente giungere. Le dimensioni dei buffer di acquisizione determinano il valore massimo della latenza ammessa. In ogni caso una latenza controllabile, necessaria per l'impiego in tempo reale, può essere difficilmente ottenuta in un PC che non sia opportunamente preparato allo scopo.

Le componenti che maggiormente introducono fluttuazioni e aumenti della latenza in un PC standard sono dovute alla gestione non deterministica della CPU della comunica-

zione tra l'interfacce (NIC e GPU) con la RAM.

Per ovviare al problema si studierà l'applicazione contestuale delle seguenti metodologie:

- driver di rete per la riduzione della latenza intrinseca del trasferimento,
- impiego di sistemi operativi realtime basati su Linux,
- metodi di trasferimento dati senza impiego di CPU (GPUDirect e sue evoluzioni)
- utilizzo di protocolli di rete per la sincronizzazione al livello del microsecondo.

I sistemi operativi realtime (RTOS) offrono la possibilità di controllare con un buon livello di accuratezza la predicibilità con cui un compito deve essere svolto. Questo apporta un miglioramento agli aspetti di non predicibilità della CPU come la gestione dei driver o il trasferimento del software di calcolo sulla GPU. Per permettere l'integrazione con GPU nell'ottica del progetto, si considereranno soluzioni commerciali e/o open-source basate sul sistema operativo Linux, quali RTLinux, RTAI, Xenomai e RedHawk.

Per minimizzare la latenza di trasferimento dei dati dalla NIC alla memoria della GPU valuteremo le tecnologie emergenti per trasferimento dati diretto verso le GPU, che l'industria sta sviluppando per permettere, ad esempio, l'acquisizione da periferiche video. Un esempio concreto è rappresentato dall'approccio GPUDirect di NVIDIA, per ridurre la latenza di trasferimento da interfacce di rete specifiche, che ha la potenzialità di svilupparsi verso il trasferimento peer-to-peer tra NIC e scheda video, evitando completamente qualunque interazione con le funzionalità dell'host.

Nell'applicazione a sistemi di trigger sincroni in HEP, la latenza complessiva della decisione di trigger deve essere costante a livello dei nanosecondi, e in caso di esperimenti esistenti la trasmissione deve essere realizzata secondo un protocollo specifico stabilito. Queste due funzioni (risincronizzazione finale e adattamento di protocollo) saranno realizzate per mezzo di una scheda elettronica interfacciata con il clock dell'esperimento. Nel caso dell'esperimento NA62 utilizzeremo lo standard TTC del CERN.

La parte centrale del progetto consiste nell'uso della GPU. La GPU è strutturata in modo da offrire molteplici livelli di parallelizzazione impiegando diverse centinaia di unità di calcolo. Per i nostri scopi intendiamo sfruttare l'architettura delle GPU per due livelli di parallelizzazione: sull'algoritmo e sugli eventi.

Per parallelizzazione sull'algoritmo si intende la possibilità di scrivere algoritmi strutturalmente paralleli, in modo da poter sfruttare il calcolo concorrente per aumentare la velocità di esecuzione per evento processato. Per parallelizzazione sugli eventi si intende la suddivisione logica delle unità di calcolo all'interno della GPU, per processare contemporaneamente più eventi. Il bilanciamento delle risorse per poter meglio sfruttare questi due aspetti del calcolo parallelo è determinante per l'ottimizzazione dei tempi di calcolo e, quindi, del rate sostenibile per l'esecuzione in tempo reale.

Per studiare applicazioni pratiche dell'impiego delle GPU prenderemo in considerazione il problema di pattern recognition nei trigger di esperimenti di alta energia, come quelle presenti negli esperimenti NA62 e ATLAS del CERN (ad esempio ricostruzione veloce di cerchi in un contatore Cerenkov, ricostruzione di tracce in un rivelatore al silicio, e ricostruzione di tracce in uno spettrometro magnetico di grandi dimensioni).

Sfruttando il sistema di trigger e acquisizione dei dati (TDAQ) di NA62 si studierà l'applicazione di quanto realizzato nel primo livello sincrono di trigger (L0).

La latenza massima a disposizione per ricevere i dati, elaborarli e trasmettere la decisione di trigger ai moduli di read-out è dell'ordine del millisecondo. La frequenza media di eventi sui rivelatori principali è dell'ordine di 10 MHz. Utilizzando le informazioni provenienti dal RICH (Ring Imaging Cerenkov) sarà possibile ricostruire con buona precisione la velocità, inferire il tipo di particella e la massa mancante del sistema. Per la ricostruzione dei cerchi nel RICH di NA62 abbiamo effettuato una valutazione preliminare su diversi modelli di GPU. I risultati pubblicati in [1], supportano la validità dell'approccio proposto. Il tempo di processamento per pacchetti da 1000 eventi, che, in alcuni casi è inferiore ai 50 ns.

L'applicazione in ATLAS punterà soprattutto all'impiego del calcolo su processori video nel livello software del trigger (LVL2). In questo contesto le richieste sulla latenza non sono stringenti come nel caso di un trigger sincrono, ma le necessità di potenza di calcolo e dimensioni degli eventi sono sicuramente più rilevanti.

Svilupperemo un framework per implementare in maniera efficiente il porting degli algoritmi attuali del trigger di LVL2 di ATLAS. Attualmente gli algoritmi sono implementati per dare risposte approssimate, che sarebbero sicuramente migliorabili avendo a disposizione più potenza di calcolo. Il disporre di algoritmi più complessi migliorerà la sensibilità dell'esperimento a segnali di fisica, offrendo soprattutto una migliore tolleranza nella ricostruzione di eventi con un alto grado di sovrapposizioni di collisioni p-p (pileup).

Questo sarà particolarmente utile in caso di aumento di luminosità, già nella seconda parte del programma di presa dati attuale e nei futuri upgrade di LHC. Si studierà la sostituzione dell'algoritmo attuale dei muoni, MuFast, con un algoritmo di tipo Kalman Filter in parallelo sulle differenti parti del rivelatore, in modo da ridurre il tempo totale di elaborazione dell'evento totale. Si studieranno quindi algoritmi di isolamento dei muoni, al momento limitati proprio dalla potenza di calcolo a disposizione. Per ultima cosa si studierà la possibilità di unire le informazioni calorimetriche e del tracciatore interno con le informazioni dei muoni, in modo da applicare selezioni più raffinate in ambiente con alto pileup e in modo da ricostruire i vertici secondari con muoni direttamente a livello di trigger.

Per mostrare i possibili impieghi del calcolo parallelo su processori video, anche fuori dall'ambito della fisica delle particelle, considereremo, in questo stesso progetto, le applicazioni nella ricostruzione di immagini in ambito medicale su macchine NMR, PET e CT. Le GPU si sono dimostrate, infatti, degli strumenti molto utili nella fisica medica a causa della sempre crescente mole di dati generata dai moderni strumenti di imaging e dalla necessità di processarli in tempi sempre più brevi. Gli studi in questo ambito si sono concentrati, principalmente, su tre possibili vantaggi che il calcolo su GPU potrebbe portare :

- La ricostruzione di immagini mediche in tempo reale può consentire agli operatori l'ottimizzazione interattiva della stessa anche rispetto ad una condizione dinamica.

1.2. Confronto con altre tecniche di processamento in tempo reale

- L'ottimizzazione dei trattamenti radioterapici in modo da migliorare l'accuratezza della distribuzione di dose somministrata ai pazienti.
- Lo sviluppo di algoritmi per l'analisi automatica di immagini mediche.

Per quanto riguarda la NMR studieremo l'applicazione, nel nostro sistema, di algoritmi su GPU utilizzando nuove tecniche diagnostiche. In particolare lo studio di alcuni parametri fisici, come la diffusione anomala non gaussiana e la kurtosis diffusionale, sono attualmente di grande interesse, perché sembrano indicare nuove strade per la diagnosi precoce di patologie cerebrali, sclerosi multipla e malattie tumorali. L'intensità di calcolo necessaria per la ricostruzione di immagini con queste tecniche è molto elevata (più di 20 ore per immagine) e questo le rende di scarso interesse diagnostico. L'impiego delle GPU porterebbe una notevole accelerazione nella velocità di ricostruzione, rendendo possibile l'impiego nella pratica clinica. Studieremo il sistema proposto direttamente su macchine NMR funzionanti: in particolare useremo una macchina NMR sperimentale per piccoli animali in dotazione al laboratorio di Risonanza Magnetica Nucleare di Roma e un apparecchio NMR total body di 3 Tesla, grazie al coinvolgimento, in questo stesso progetto, dell' UOC di Tecniche diagnostiche avanzate del Policlinico Umberto I di Roma.

La ricostruzione delle immagini provenienti da un tomografo PET verrà studiata con particolare riguardo agli algoritmi Order Subset Expectation Maximization (OSEM) e Maximum Likelihood Expectation Maximization (MLEM) in versione list-mode, particolarmente idonei alla parallelizzazione su GPU [7]. Anche in questo caso per gli scopi del progetto si disporrà di due tomografi PET sperimentali (YAP(S)PET e DoPET), grazie alla collaborazione del CNR e del gruppo di Fisica medica dell'INFN di Pisa.

Nel contesto dell'imaging CT ci occuperemo di adattare algoritmi iterativi al calcolo su GPU, come Expectation Minimization (EM), OSEM o Simultaneous Iterative Reconstruction Technique (SIRT), in modo da permettere una ricostruzione realtime capace di seguire il flusso delle immagini acquisite. Ci dedicheremo in particolare alla moderna tecnica del Cone Beam Computerized Tomography (CBCT) che permette di limitare il numero di proiezioni da acquisire e quindi la dose al paziente. Per far questo sfrutteremo la collaborazione con la sezione di Medicina Nucleare del Dipartimento di scienze Chirurgiche Anestesiologiche e Radiologiche dell'azienda Ospedaliera Universitaria di Ferrara e del Laboratorio Raggi-X dell'Università.

1.2 Confronto con altre tecniche di processamento in tempo reale

Non è questa la sede per discutere in dettaglio le varie tecniche che si impiegano o si potrebbero impiegare per il processamento in tempo reale, e, in particolare, delle varie "filosofie" per implementare trigger in esperimenti HEP, con bassa latenza e alto throughput di calcolo. Giusto per completare la discussione sui motivi del progetto GAP, ci sembra opportuno comunque fare un cenno a tale confronto in modo da meglio comprendere perchè riteniamo interessante caratterizzare nelle applicazioni proposte dei processori originariamente pensati per scopi diversi.

La struttura dei trigger di basso livello disegnato fino ad oggi è stata condizionata soprattutto dalla bassa latenza, causata dall'impossibilità di disporre di memorie capienti a livello dell'elettronica di frontend. Unitamente alla specificità dei problemi tipici delle applicazioni HEP, questo ha portato allo sviluppo di elettronica veloce e dedicata, rinunciando spesso alla versatilità e alla scalabilità. Le necessità imposte dalla alta banda e dall'alto rate, hanno imposto il fatto che la decisione di trigger fosse basata solo su un set ridotto dei dati da processare.

Negli ultimi anni alcune di queste limitazioni sono state superate dalle innovazioni tecniche che sono avvenute soprattutto nell'elettronica di consumo al di fuori della ambito strettamente legato alla fisica delle alte energie.

Per prima cosa, a causa delle richieste delle reti commutate veloci, si sono sviluppate strutture di trasporti dei dati estremamente veloci e affidabili. In secondo luogo il costo e l'integrazione delle memorie sono diventati talmente competitivi da permettere di pensare a sostanziali incrementi delle capacità di bufferizzazione di dati digitali.

Questi argomenti insieme al notevole aumento della capacità di calcolo dei PC, ha portato ad iniziare a pensare ad architettura di trigger in cui la componente hardware e custom viene sempre più ridotta, puntando idealmente ad architetture triggerless (strutture ad altissima banda di trasporto) oppure in cui la decisione di trigger venga presa su dati più complessi (strutture con grande capacità di calcolo online). Al momento architetture completamente triggerless, in cui i dati vengano completamente spostati su PC per prendere una decisione veloce online, appare difficile da realizzare a causa delle sempre più importanti quantità di dati prodotte dai grandi esperimenti e dai costi delle infrastrutture di rete.

Per quanto riguarda il calcolo, invece, diverse soluzioni si stanno considerando. Molto semplificando, nello scopo di queste note, la decisione di trigger di livello basso si può basare sia sul calcolo di alcune grandezze fisiche a partire dagli hit grezzi raccolti dai rivelatori, oppure dal confronto veloce con pattern noti. Il primo metodo richiede elaborazione veloce di informazioni, il secondo tecniche di tipo "memoria associativa" per confrontare con il più alto numero di configurazioni possibili. E' da notare che il secondo metodo si è sviluppato soprattutto a causa dell'incapacità, nel passato, di ottenere da processori standard le performance richieste dalle applicazioni tipiche degli esperimenti di fisica delle alte energie.

Negli ultimi anni, però, le FPGA e, in modo differente, i processori hanno dimostrato di poter offrire throughput di calcolo estremamente competitivo, per permettere un design estremamente più versatile e scalabile rispetto a soluzioni custom. Tale sviluppo nell'elettronica di consumo non è stato stimolato direttamente dalle richieste del settore di ricerca in fisica delle alte energie, quanto dalle richieste di settori commercialmente più rilevanti, come le reti e l'high performance computing per scopi differenti. Tra i processori, nell'ultimo periodo, le architetture parallele si stanno imponendo nella definizione del nuovo standard di potenza di calcolo e in questo ambito le GPU dominano il settore.

Lo scopo del progetto GAP è proprio quello di capire i vantaggi e le limitazioni dei processori grafici nel settore dei trigger, in un contesto in cui lo sviluppo continuo di questi device garantisce potenze di calcolo e performance difficilmente ottenibili dall'elettronica custom, come è invece successo fino ad ora.

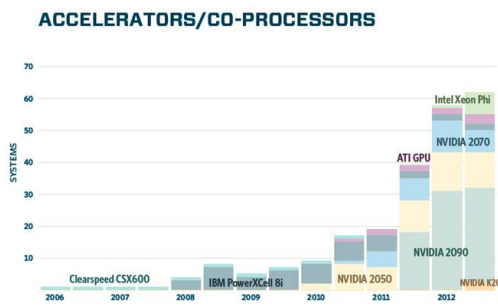


Figura 1.2: Numero di Super Computer nei primi 500 più potenti, basati su tecnologie di calcolo parallelo con coprocessori.

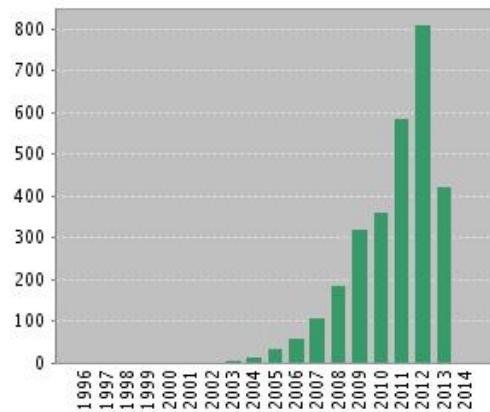


Figura 1.3: Esempio di andamento di articoli scientifici che parlano di “GPU computing” (fonte ISIWeb)

1.3 Stato dell'arte delle GPU

Anche questo paragrafo non vuole essere esaustivo di un argomento estremamente vasto e, comunque, parzialmente non rispondente allo scopo di questo report. Vogliamo soltanto introdurre lo stato dell'arte dei moderni processori grafici con lo scopo di meglio spiegare le scelte effettuate in questo progetto, rimandando a più autorevoli fonti (per esempio [2]) discussioni più approfondite sull'argomento.

L'impiego dei processori grafici per calcolo scientifico ha avuto un enorme incremento a partire dal 2007, ovvero da quando la NVIDIA e la ATI (ora AMD) hanno reso possibile un semplice accesso alle potenzialità computazionali delle GPU. Contestualmente le richieste dell'industria dell'entertainment hanno sempre di più spinto lo sviluppo di processori potenti. Dal lato connessione con il PC, per garantire una connessione sempre più di alta banda, si è sviluppato un nuovo standard, il PCI express, che, giunto alla versione 3, permette di limitare il più possibile le limitazioni imposte dalla necessità di spostare i dati sulla scheda video quando impiegata come coprocessore matematico.

Nel grafico di figura 1.2 si può vedere come nei moderni supercomputer l'impiego di processori video per aumentarne la capacità di calcolo stia diventando sempre più comune, al punto che il computer più potente ad oggi, è basato proprio sull'impiego di questa tecnologia. Nel grafico 1.3 è invece evidente come stiano notevolmente aumentando gli articoli scientifici (nell'esempio il grafico è ottenuto ricercando “GPU computing” su ISI-Web) che impiegano le GPU per scopi non strettamente legati alle loro capacità di gestire la grafica.

D'altra parte negli ultimi anni si è molto sviluppata la necessità di disporre di enorme capacità di calcolo in singole macchine facilmente gestibili anche da programmatori poco esperti, e questo ha portato anche a studiare soluzioni differenti basate sull'impiego di gruppi di processori tipo X86, processori ARM, ibridi CPU-GPU o schede multi-FPGA. Il risultato è che è molto difficile stabilire una metrica oggettiva per confrontare le differenti soluzioni. In ogni caso le GPU offrono diversi vantaggi che vanno dalla potenza di calcolo per watt consumato, alla facile gestione, al costo ridotto, al continuo sviluppo. Nella

Tabella 1.1: Caratteristiche delle più recenti GPU dedicate al calcolo.

Video Card	n. of cores	Processing Power (GFLOPS)	Memory Bandwidth (GB/s)
NVIDIA TESLA C1060 (2009)	240	933	102.4
NVIDIA TESLA C2050 (2011)	448	1288	144
NVIDIA TESLA K20 (2013)	2496	3520	208
NVIDIA GFORCE GTX680 (2012)	1536	3090	192
AMD RADEON 5870 (2010)	1600	2720	153.6
AMD RADEON 7970 (2012)	2048	4096	288

tabella 1.1 riportiamo le caratteristiche delle ultime GPU disponibili sul mercato.

In questi giorni, mentre scriviamo questo report, è stata annunciata, da NVIDIA, la nuova scheda K40 che promette un nuovo standard in fatto di potenza di calcolo con maggiore di 4 TFlops e una banda con la memoria di quasi 300 GB/s. Questo a riprova del fatto che la potenza di calcolo disponibile nelle schede video, ha una crescita estremamente elevata non essendo vincolata alle limitazioni imposte dalla saturazione della legge di Moore, considerando che l'aumento di potenza non è direttamente connesso alla velocità di clock e all'integrazione spinta, quanto piuttosto alle dimensioni del chip.

1.4 Profilo economico del progetto

Concludiamo questa sezione del report discutendo il finanziamento concesso al progetto e la sua suddivisione iniziale tra le unità di ricerca.

I progetti di tipo FIRB vengono finanziati in misura del 70% del costo congruo con eccezione degli stipendi per giovani ricercatori che vengono pagati interamente dal MIUR. La quota mancante al costo del progetto va prodotta esponendo lo stipendio di ricercatori dipendenti coinvolti nella ricerca, nella misura massima del 30%. Se non si satura la parte del 30% il rimanente può essere pagata attingendo dalle spese generali che sono generate in quota forfettaria rispetto a tutti i costi del personale.

Nella proposta del progetto abbiamo presentato un budget con un costo congruo totale di 1046198 euro (con un finanziamento MIUR effettivo richiesto di 821439). In fase di approvazione il finanziamento concesso è stato tagliato in due fasi, dalla commissione esaminatrice e da questioni legate al budget totale del 2012, e portato ad un costo congruo effettivo di 836620 euro con un finanziamento concesso di 648004 euro.

Tale finanziamento è stato rimodulato tra le tre unità come mostrato nella tabella 1.2. Nella tabella 1.3 riportiamo la suddivisione dei fondi effettivi concessi all'unità INFN. In questa tabella non è riportato il budget concesso per lo stipendio del Principal Investigator che, con alcuni vincoli, ritornerebbe a disposizione dell'unità in caso di assunzione a

INFN(Pisa)	340494
Ferrara	281192
Roma	214934
Totale	836620

Tabella 1.2:

Consumo	92257
Missioni	8982
Altro	2000
Totale	103239

Tabella 1.3:

1.4. Profilo economico del progetto

tempo indeterminato nell'ente di afferenza.

Il progetto è stato presentato dai proponenti (Gianluca Lamanna, Massimiliano Fiorini e Andrea Messina) a fine Febbraio 2012. A Giugno ha passato la preselezione dell'INFN e a metà ottobre quella dei referee internazionali nominati dal ministero. A fine ottobre stato svolto l'audit, e l'approvazione al finanziamento è stata data a metà novembre. Il progetto è iniziato ufficialmente il 1.4.2013.

Capitolo 2

Unità di Ricerca e collaborazioni

Il progetto GAP è portato avanti da tre unità di Ricerca:

- Unità INFN (Pisa): Gianluca Lamanna (Principal Investigator)
- Unità di Ferrara: Massimiliano Fiorini
- Unità di Roma “La Sapienza”: Andrea Messina

Ognuna delle tre unità è virtualmente divisa in due macro-aree di interesse per lo studio delle applicazioni in trigger di HEP e lo studio della ricostruzione delle immagini in applicazioni medicali. I compiti delle unità sono divisi a seconda delle caratteristiche e degli interessi dei ricercatori partecipanti. Inoltre, considerando il preesistente coinvolgimento dei responsabili in esperimenti HEP, i compiti sono divisi in modo da integrare le attività di sviluppo in frame di studio esistenti.

In fig.2.1 si riporta uno schema delle attività previste nei vari anni del progetto per ogni unità. Nel seguito si riportano i dettagli e l'organizzazione delle varie unità di ricerca, evidenziando anche le collaborazioni e i collegamenti con altri gruppi e persone non direttamente afferenti al progetto.

2.1 Unità INFN di Pisa

L'unità di Pisa è costituita, da progetto, da:

- Gianluca Lamanna: coordinatore dell'unità e PI del progetto
- Niccolò Camarlinghi,
- Flavio Costantini,
- Gianmaria Collazuol,
- Riccardo Fantechi,
- Mauro Piccini,
- Marco Sozzi .

G.Collazuol è dipendente presso l'Università di Padova e M.Piccini è dipendente INFN presso la sezione di Perugia, tutti gli altri collaboratori sono a vario titolo dipendenti dell'Università o dell'INFN di Pisa. Oltre i ricercatori sopra citati, afferenti direttamente a GAP, altri studiosi collaborano attivamente all'attività dell'unità. In particolare Jacopo Pinzino (dottorando) e Roberto Piandani (assegnista) collaborano su diversi aspetti della ricerca sui processori grafici. L'unità sta anche beneficiando del contributo di due laureandi (Elena Graverini e Felice Pantaleo). Inoltre è stato instaurato un rapporto di collaborazione con ricercatori del dipartimento di informatica (Luca Deri, Alfredo Carmigliano, Gabriele Cocco) e, in modo estremamente rilevante per gli obiettivi dell'unità, con il gruppo Apenet di Roma "Sapienza", nelle persone di Piero Vicini, Alessandro Lonardo, Andrea Biagioni, Roberto Ammendola, Francesco Simula e altri.

Come nel caso delle altre unità, l'attività dell'unità INFN di Pisa è divisa in due parti distinte:

- applicazioni delle GPU in bassi livelli di trigger
- studio della ricostruzione di immagini PET.

Durante la riunione del 31.5.2013 a Bologna sono stati presentati ai revisori del progetto i seguenti obiettivi a breve termine:

- Caratterizzazione delle due soluzioni di trasporto dati (PFRING e NANET) in un sistema a scala ridotta del sistema di trigger e acquisizione di NA62.
- Misura della latenza
- Algoritmi sui Cerchi (L0)

	I ANNO Realizzazione	II ANNO Caratterizzazione	III ANNO Integrazione
PISA	<ul style="list-style-type: none"> • Gestione della latenza (10, 40 Gb/s) • Algoritmi per Ring • Algoritmi list-mode per PET 	<ul style="list-style-type: none"> • Misura della latenza di trasferimento in varie modalità • Studio dell'integrazione in NA62 • Studio dell'integrazione con PET 	<ul style="list-style-type: none"> • Integrazione in NA62 • Misure di efficienza in NA62 • Integrazione in PET e studio di performance e vantaggi
FERRARA	<ul style="list-style-type: none"> • Sviluppo UNICA • Algoritmi per GTK • Algoritmi per CT (FDK iterativo) 	<ul style="list-style-type: none"> • Caratt. UNICA su sistemi sincroni • Algoritmi per GTK • Algoritmi SIRT e EM e studio per integrazione con CT 	<ul style="list-style-type: none"> • Integrazione in NA62 • Misure di efficienza in NA62 • Integrazione in CT e studio di performance e vantaggi
ROMA	<ul style="list-style-type: none"> • Sistemi RTOS • Algoritmi per mu • Algoritmi per NMR (kurtosis diffusionale) 	<ul style="list-style-type: none"> • Integrazione dei sistemi di riduzione della latenza in sistemi RTOS • Studio integrazione con NMR 	<ul style="list-style-type: none"> • Integrazione in ATLAS • Analisi di dati veri e/o simulati • Integrazione in NMR e studio di performance e vantaggi

Figura 2.1: Riassunto dei compiti delle unità.

- Versione definitiva della versione degli algoritmi dei cerchi multipli
- Studio dei vantaggi portati dall'uso del RICH nel L0 di NA62
- Algoritmi sullo spettrometro (L1 o L0)
- Confronto con Ferrara per stabilire i punti comuni degli algoritmi PET-CT

Per quanto riguarda le applicazioni nei bassi livelli di trigger lo studio si sta concentrando essenzialmente sul confronto delle due soluzioni per la riduzione della latenza, come descritto nel par.4.2 . Tali studi sono effettuati con la collaborazione, sopracitata, di alcuni ricercatori del dipartimento di informatica per quanto riguarda la soluzione che si basa sull'uso del driver ottimizzato per la cattura veloce e del gruppo Apenet di Roma per quanto riguarda la soluzione che si basa sul trasferimento diretto dei dati dalla scheda di ricezione alla GPU. Si prevede anche di studiare la latenza in un sistema costituito da un processore ibrido CPU-GPU. Per quanto riguarda gli studi su algoritmi paralleli da implementare per la ricostruzione di immagini PET, l'attività, al momento, è stata limitata a studi preliminari e a discussioni con l'unità di Ferrara in merito a possibili strategie comuni.

2.2 Unità di Ferrara

L'unità di Ferrara è costituita da:

- Massimiliano Fiorini: responsabile dell'unità
- Stefano Chiozzi
- Angelo Cotta Ramusino
- Marco Corvo
- Giovanni Di Domenico
- Alberto Gianoli
- Luciano Milano
- Guido Zavattini

In particolare Marco Corvo è stato assunto nell'ambito del progetto GAP con un assegno di ricerca triennale. Questo è stato reso possibile dal fatto che il responsabile dell'Unità è stato assunto dall'Università di Ferrara prima dell'inizio del progetto, e questo ha permesso di utilizzare i fondi destinati al suo stipendio per la stipula di assegni di ricerca triennali. A questo proposito un altro assegno triennale è stato bandito recentemente. Questo aumento di personale ha permesso al responsabile di Unità di estendere le finalità del progetto allo studio di un sistema di trigger di alto livello per l'esperimento LHCb del CERN, di cui è diventato membro ad inizio 2013.

Anche nel caso dell'unità di Ferrara l'attività nell'ambito del progetto è divisa in due parti:

- applicazioni delle GPU nei trigger
- studio della ricostruzione di immagini CT.

Per quanto riguarda il primo obiettivo l'unità si occuperà delle applicazioni delle GPU in due ambiti: nel contesto dei trigger di livello più basso si occuperà dello sviluppo, in collaborazione con l'unità INFN e il gruppo Apenet, dello sviluppo della scheda per il ricevimento dei dati e il trasferimento diretto alla GPU, in particolare per quanto riguarda gli aspetti di integrazione con l'esperimento NA62. Sempre per quanto concerne i trigger di HEP, l'unità di Ferrara studia, in collaborazione con il gruppo LHCb di Padova, le possibili applicazioni delle GPU nel trigger di alto livello di LHCb.

Per questo primo periodo, come indicato ai referee nell'incontro del 31.5.2013, l'unità di Ferrara ha indicato i seguenti obiettivi a breve termine:

- Definizione e disegno della scheda daughter per la ricezione del clock e il collegamento con la struttura di distribuzione del trigger e clock di NA62.
- Definizione dell'interfaccia con firmware NANET (in particolare interfaccia con il modulo di preprocessamento).
- Definizione della gestione del trasferimento di rete (offload del protocollo, 1 Gb/s, 10 Gb/s, etc..)
- Studio di meccanismi di DMA tra FPGA e RAM del PC
- Pattern recognition per tracciatori al silicio, studio preliminare e applicazione al GTK di NA62.
- Implementazione di algoritmo FDK per GPU
- Sviluppo di algoritmo con proiettore e retroproiettore veloce
- Studio di diverse famiglie di proiettori
- Studio dell'ottimizzazione in funzione della dimensionalità
- Codice di simulazione MC su GPU

A questi obiettivi indicati si la collaborazione, almeno per la fase iniziale, con il gruppo LHCb di Padova per l'utilizzo delle GPU nelle trigger di alto livello. In particolare in questo momento si sta studiando il framework online di LHCb. L'unità di Ferrara si occuperà in generale di studiare algoritmi, in particolare di tracking, idonei ad essere impiegati in trigger di alto livello sia di LHCb che di NA62.

2.3 Unità di Roma

L'unità di Roma è costituita da:

- Andrea Messina: responsabile dell'unità
- Matteo Bauce

- Silvia Capuani
- Stefano Giagu
- Andrea Laghi
- Marco Palombo
- Marco Rescigno

e si occupa di due ambiti distinti

- applicazioni delle GPU in HLT
- uso delle GPU nella ricostruzione delle NMR

Matteo Bauce è stato assunto nell'ambito del progetto con un assegno di ricerca di 1 anno con possibilità di rinnovo.

L'unità di Roma include tra i suoi partecipanti diversi membri coinvolti nell'analisi e nell'upgrade di Atlas. Per questa ragione, per quanto riguarda l'attività sui trigger HEP, l'unità si occupa essenzialmente di studiare algoritmi per il trigger di muoni di Atlas in versione tale da sfruttare le caratteristiche delle GPU. Dal punto di vista della ricostruzione delle immagini per NMR, l'unità si avvale della collaborazione del laboratorio NMR dell'università e del policlinico Umberto I. Alcuni studi preliminari sugli algoritmi di ricostruzione in nuove modalità di acquisizione con un alto potenziale diagnostico, sono stati effettuati.

Gli obiettivi da raggiungere nel primo anno di attività dell'unità di Roma sono descritti nel modello di richiesta di finanziamento. Tali obiettivi, rimasti sostanzialmente invariati, sono stati discussi con tutti i membri del progetto durante la prima riunione nazionale della collaborazione GAP il 10.5.2013 e presentati a revisori INFN nella riunione 31.05.2013.

Per comodità gli obiettivi principali vengono riassunti di seguito:

1. Acquisizione, configurazione, mantenimento di un server con scheda Nvidia GTX Titan su cui sviluppare gli algoritmi HLT e NMR e da mettere a disposizione della collaborazione GAP.
2. Identificazione in ATLAS di algoritmi di trigger di secondo livello adatti ad essere eseguiti su architetture parallele.
3. Sviluppo di primi algoritmi di test elaborati sulle librerie CUDA.
4. Interazione ed interfaccia tra il software di ATLAS e le librerie CUDA.
5. Studio e caratterizzazione dei diversi sistemi operativi con driver grafici.
6. Realizzazione di mappe in diffusione non gaussiana a livello cerebrale di soggetti già acquisiti.

In aggiunta a quanto sopra, in collaborazione con l'unità di Pisa, l'unità di Roma ha iniziato ad occuparsi della caratterizzazione di sistemi operativi realtime. Per il momento è stato acquistato dall'unità l'hardware necessario per capire quale sistema operativo basato su linux RTOS sia il più idoneo a funzionare con i driver invidia, o in generale, il migliore a gestire calcolo su processori grafici.

2.4 Riunioni periodiche e riunioni nazionali

L'organizzazione dell'attività è affidata localmente ai responsabili delle varie unità, che si occupano sia degli aspetti scientifici che amministrativi. I responsabili si sentono frequentemente in riunioni telematiche con frequenza normalmente bi-settimanale. Ogni 6 mesi si prevede di organizzare una riunione nazionale in cui tutti i collaboratori sono invitati a partecipare. La prima riunione (fig.2.2 di questo tipo si è svolta a Roma il 10.5.2013 e ha visto coinvolte almeno 25 persone provenienti dalle tre unità, con 17 talks totali 2.3. In questa riunione sono state definite le attività per i successivi 6 mesi, focalizzando soprattutto l'attenzione sulle necessità dei vari gruppi in termini di hardware e start up. La successiva riunione si svolgerà a Pisa all'inizio di Gennaio.

Il sito del progetto (<http://web2.infn.it/gap>) diventerà punto di riferimento per lo scambio di informazioni e la presentazione dei risultati, ma al momento ancora non è molto sviluppato (fig.2.4).

Tutti i collaboratori (38 persone in totale) sono inclusi in una mailing list e ricevono con cadenza bi-settimanale una newsletter di aggiornamento sulle attività che si svolgono e su notizie di interesse generale (fig.2.5).



Figura 2.2:

I meeting generale GAP (Roma)
shared by Gianluca Lamanna (PD), Andrea Meazza (ROMA1), Massimiliano Fiorini (CERN)
Friday, 10 May 2013 from 11:00 to 19:00 (Europe/Rome)
at Roma - La Sapienza - ed.Marconi (Aula presidenza INFN)
Roma University La Sapienza Via. Di Piazzale Palazzi 950 Roma, 5

Description Primo meeting generale GAP
 Il meeting coinvolge le tre unità di ricerca di Pisa, Roma e Ferrara e sarà diviso in due parti: la mattina dedicata a discussioni riguardanti il livello 1 e il pomeriggio al livello 1 e fisica medica, oltre che programmazione degli obiettivi e questioni organizzative varie.

Friday, 10 May 2013

11:00 - 13:00 **Discussione sul L0**
 - introduzione
 - note
 - PF_RING
 - AME2
 - interazione femra-appenet
 - sistemi ATLAS
 - sviluppi futuri

11:00 **Introduzione 20'**
 Speaker: Gianluca Lamanna (PI)
 Materiali: [Slides](#) [v](#)

11:20 **RANET 20'**
 Speaker: Alessandro Lonardo (ROMA1)
 Materiali: [Slides](#) [v](#)

11:40 **Sviluppo scheda UNICA 20'**
 Speaker: Massimiliano Fiorini (CERN)
 Materiali: [Slides](#) [v](#)

12:00 **PF_RING 20'**
 Speaker: Alfredo Cardigliano (PI)
 Materiali: [Slides](#) [v](#)

12:20 **Sistemi operativi Realtime 20'**
 Speaker: Mauro Piccini (PD)
 Materiali: [Slides](#) [v](#) [v](#)

13:00 - 14:00 **Pranzo** [...]

14:00 - 14:30 **Discussione su Livelli software**
 - idee su ATLAS
 - idee su AME2

14:00 **Introduzione HLT ATLAS 20'**
 Speaker: Andrea Messina (ROMA1)
 Materiali: [Slides](#) [v](#) [v](#)

Figura 2.3:

GAP
Realtime for HEP and Medical Imaging

HOME INDICIO GAP LISTA GAP

Set sul Home

MENU PRINCIPALE

HOME
 CHI SIAMO
 TRIGGER & GPU
 FISICA MEDICA & GPU
 OPPORTUNITÀ DI LAVORO IN GAP
 NEWS FEED
 CERCA
 LINK WEB
 CONTATTI
 ACCESSO UTENTI
 AMMINISTRAZIONE
 TALK A CONFERENZE

Intro
 GAP (GPU applications project) - Realtime for HEP and Medical Imaging è un progetto finanziato nell'ambito del PRIS-2012 "Futuro in Ricerca" a cui collaborano tre unità di ricerca: INFN (sez. di Pisa), Università di Roma "Sapienza" e Università di Ferrara. Lo scopo principale del progetto è dimostrare l'efficacia dei processori della serie schede grafiche (GPU) in applicazioni scientifiche in tempo reale, in particolare nelle tre unità di ricerca si studiano simultaneamente applicazioni in:

- Trigger per esperimenti di fisica delle alte energie
- Ricostruzione di immagini in scanner medici

Per quanto riguarda i sistemi di trigger in esperimenti di fisica delle alte energie, si considerano sia le applicazioni in livelli di trigger software, dove l'utilizzo di processori grafici può essere di notevole aiuto per aumentare le capacità di calcolo online, sia applicazioni in livelli hardware, dove le tecniche di risposta di elaborazione e lettura sono più stringenti. Gli esperimenti ATLAS e CMS vengono considerati, rispettivamente, come physics case per tali studi.

Per quanto riguarda la ricostruzione di immagini mediche, il progetto si occuperà di ricostruzione di immagini per PET, CT e MRI. Grazie alla collaborazione con unità di ricerca e ospedali, il progetto potrà avvalersi di scanner e immagini acquisiti in contesti reali.

Figura 2.4:

Sito GAP Forward

GAP NewsLetter

TABLE OF CONTENTS

Presentazione & Comunicazioni

- [Presentazione](#)
- [Personale](#)
- [PC](#)
- [acquistati](#)
- [Conferenze](#)
- [Attività](#)
- [Licenze](#)
- [Altre](#)
- [news](#)

Al fine di tenervi aggiornati su progressi, attività e novità abbiamo pensato a questa "newsletter" con cadenza, sperabilmente, settimanale (ogni lunedì). Cercheremo di mettere le informazioni rilevanti per quanto riguarda quello che si fa nelle varie unità di ricerca e altre notizie che potrebbero essere di interesse generale (segnalazione articoli, contatti, conferenze, etc.). Tutti possono contribuire con report, considerazioni, informazioni o link. Per meglio organizzare la cosa vi chiedo di inviarmi i contributi "lungini" con un paio di giorni di anticipo (le cose brevi possono anche essere inviate "all'ultimo minuto"). Questa prima "edizione" è sperimentale, nel senso

Figura 2.5:

Capitolo 3

Acquisti e Personale

La prima fase del progetto è stata caratterizzata soprattutto dagli acquisti dei PC necessari ad effettuare gli studi proposti. Alcuni di questi PC erano già disponibili nell'unità di Pisa, mentre altri sono stati acquistati in modo da utilizzare le ultime versioni dei processori video. Al fine di ottimizzare la spesa è stato deciso di acquistare, per gli scopi di studi di trigger di alto livello e di ricostruzione di immagini medicali, un PC particolarmente potente (Server Supermicro doppio processore Xeon con GTX TITAN), ospitato nell'unità di Roma al quale si connettono remotamente anche i ricercatori di Ferrara e Pisa. Gli altri PC acquistati nelle altre unità, sono di costo decisamente inferiore con lo scopo di permettere uno sviluppo locale e i test hardware descritti nel capitolo del L0. Inoltre sono stati acquistati appositamente un PC per lo studio dei sistemi operativi realtime (unità di Roma) e un PC per ospitare un processore ibrido AMD CPU-GPU e studiarne le caratteristiche (unità di Pisa). Insieme a questa tipologia di spesa, altro capitolo rilevante è stato quello legato all'assunzione di personale per gli scopi del progetto. Nel dettaglio, al momento, il personale è stato preso nelle sezioni di Roma (Matteo Bauce, assegno di ricerca annuale) e Ferrara (Marco Corvo, assegno di ricerca triennale). Nell'unità di Roma è stata anche data una borsa trimestrale per il settore di ricostruzione delle immagini NMR. Al momento di stesura di questo report l'unità di Ferrara è in procinto di chiudere un bando per un altro assegno triennale con profilo di elettronico.

Nella tabella successiva si riportano le spese effettuate al 13/11/2013:

	INFN(Pisa)	Roma	Ferrara	Totale
Consumo	4793	11929	0	16722
Inventariabile	0	1561	5000	6561
Missioni	3359	533	900	4792
Personale	0	30070	148000	178070
Altro	0	0	0	0
Totale	8152	44093	153900	

L'unità di Pisa ha acquistato 3 PC: con NVIDIA GTX650, con NVIDIA TITAN e con processore ibrido AMD (tipo APU). Le spese per le missioni si riferiscono alle conferenze (Amburgo e impegno per conferenza RICH a Tokio del prossimo Dicembre), alle spese per la partecipazione alla riunione nazionale e alle spese per le trasferte del PI in visita alle unità di Ferrara e Roma.

Acquisti e Personale

L'unità di Roma ha acquistato 4 PC: un server Xeon con scheda TITAN, 2 desktop con GTX650 per impiego di ufficio e un computer, con scheda GTX680, per lo studio dei sistemi operativi real time. Le missioni sono state limitate ad una trasferta per il workshop di Amburgo. Le spese per il personale si riferiscono ad un assegno di ricerca annuale e ad una borsa di studio breve.

L'unità di Ferrara ha acquistato 2 PC: uno con scheda TITAN e l'altro con GTX650. Le spese per le missioni si riferiscono alla spesa per il workshop di Amburgo e alle trasferte per la riunione nazionale. Le spese per il personale si riferiscono a 2 assegni di ricerca triennali; il concorso per l'assegnazione del secondo, al momento della scrittura di questo report, è in fase di chiusura.

Capitolo 4

GPU e L0: stato

Lo studio relativo all'impiego delle GPU a L0 è sicuramente la parte più challenging dell'intero progetto. Come spiegato precedentemente le schede grafiche non sono progettate per lavorare in contesti in cui la latenza è particolarmente ridotta (hard realtime) e in cui deve essere particolarmente stabile.

Per quanto concerne gli sviluppi relativi al L0, per i primi 6 mesi del progetto abbiamo indicato i seguenti obiettivi:

- Sviluppo di due soluzioni per la riduzione delle latenze (PFRING e NaNet) per il trasporto dei dati alla GPU e misura delle performance (Pisa+Roma Apenet).
- Algoritmi per la ricerca di singolo cerchio e di cerchi multipli (Pisa).
- Studio dei sistemi operativi realtime (Roma + Pisa).
- Studio dei meccanismi di DMA tra FPGA e RAM del PC (Ferrara).
- Disegno della scheda di interfaccia con il sistema di distribuzione del trigger di NA62 (Ferrara).

Nelle successive sezioni si discuteranno i risultati e lo stato del lavoro. Il grado di completamento di molti degli obiettivi è discreto. In particolare per quanto riguarda la parte principale, la caratterizzazione dei metodi di riduzione della latenza, rimane da fare soltanto un confronto critico tra i due metodi e lo studio della dipendenza dei metodi proposti dalle architetture specifiche dei PC utilizzate.

Per quanto riguarda lo studio dei sistemi operativi realtime, al momento sono state effettuate delle indagini preliminari ed è stato acquistato, su fondi dell'unità di Roma, il PC idoneo per effettuare gli studi. Per quanto riguarda la scheda di interfaccia con il sistema di distribuzione del trigger di NA62, ci sono state delle riunioni per definire un disegno concettuale e l'interfaccia con la struttura della scheda NaNet, nei prossimi mesi verrà completato il disegno elettronico.

4.1 Ricostruzione di anelli nel contatore RICH

Abbiamo effettuato alcuni studi preliminari sulla ricostruzione online di anelli Cherenkov prendendo come case study il rivelatore RICH di NA62 [3]. La prima parte dello studio è

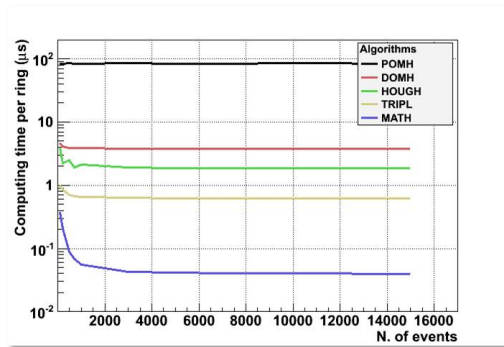


Figura 4.1: Confronto di diversi algoritmi di ricerca di singolo cerchio su diverse scheda grafica NVIDIA C1060.

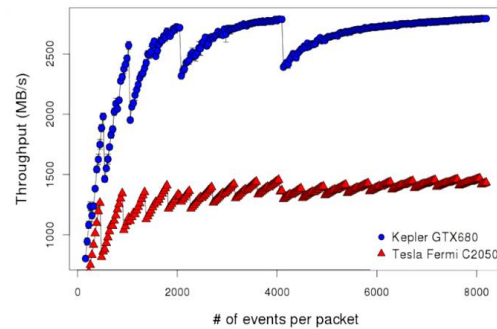


Figura 4.2: Confronto tra scheda NVIDIA C2050 e GTX680 per algoritmo MATH.

stata rivolta ad analizzare le migliori configurazioni architetturali possibili per la GPU rispetto ad alcuni algoritmi di ricerca di singoli cerchi, senza l'utilizzo di semi. Nel grafico 4.1 mostriamo i risultati ottenuti per alcuni di questi algoritmi (descritti in dettaglio in [1]) su una scheda NVIDIA TESLA C1060. Le performance di questo algoritmo sono state confrontate anche sulle schede NVIDIA TESLA C2050 e NVIDIA GTX680, di generazioni successive rispetto alla C1060 (in particolare il processore della GTX680 è lo stesso presente nelle schede K20 utilizzate per gli studi di latenza di cui si discuterà nel seguito). Come si vede nei grafici di fig.4.2 i risultati ottenuti indicano un fattore di miglioramento almeno due, in termini di throughput di calcolo e latenza totale, in generazioni successive di processori grafici.

Questi e altri risultati sono mostrati più in dettaglio in nei proceedings delle conferenze discusse al cap.7.

Il passo ulteriore è stato quello di provare algoritmi per la ricostruzione di cerchi multipli. Tra i vari algoritmi presenti sul mercato per la ricostruzione di più di un cerchio nessuno è risultato idoneo alla parallelizzazione su GPU in un contensto standalone. Infatti molti di questi algoritmi si basano per iniziare la procedura di ricerca, sulle informazioni provenienti da altri rivelatori (come ad esempio uno spettrometro) che, ovviamente, non sarebbero disponibili in un trigger di livello basso. Anche altre procedure sono intrinsecamente seriali e quindi, a priori, difficilmente potrebbero sfruttare i vantaggi offerti dal calcolo parallelo. Per affrontare questo problema è stato sviluppato un nuovo algoritmo basato sul teorema di Tolomeo. Tale teorema afferma che un quadrilatero è ciclico (ovvero può essere inscritto in un cerchio) se è solo se la somma dei prodotti dei lati opposti è uguale al prodotto delle diagonali.

L'idea alla base dell'algoritmo in questione è quella di accumulare, a partire da alcune triplette iniziali scelte con considerazioni geometriche, tutti i punti che appartengono ad un cerchio, ovvero che soddisfano la condizione di ciclicità rispetto ai tre punti scelti all'inizio, prima di effettuare un fit di singolo cerchio. Il fatto che la GPU possa calcolare contemporaneamente più triplette assicura che, con una certa efficienza, vengano risolti tutti i cerchi presenti. Ovviamente tale efficienza dipende fortemente dal numero di triplette iniziali, e quindi i tempi di calcolo, che dipendono dalle risorse impiegate, sono inversamente proporzionali alle triplette richieste. Studi preliminari indicano che, per un numero ragionevole di triplette tali da mantenere un'efficienza più alta del 90%, i tempi

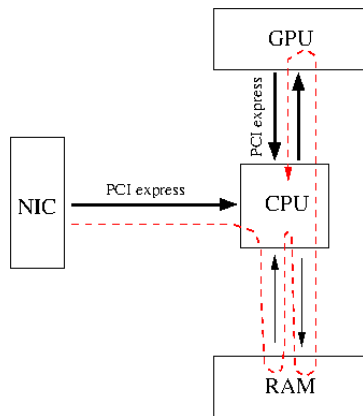


Figura 4.3: Vista schematica della procedura di trasporto dei dati in un PC.

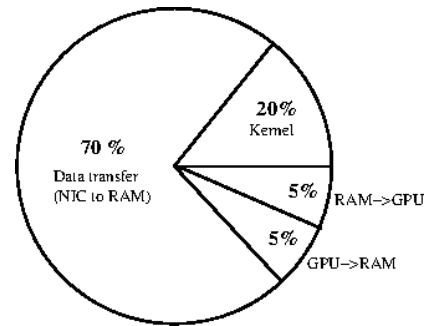


Figura 4.4: Time budget del procesamiento su GPU (assumendo un kernel relativamente semplice).

di esecuzione sono dell'ordine di qualche microsecondo su una TESLA K20. Tali tempi sono assolutamente compatibili con le richieste di NA62 in termini di rate di eventi, e necessitano comunque di possibile ulteriore ottimizzazione.

In collaborazione con l'unità di Pisa sono state svolte due tesi di laurea (Elena Grave-rini e Felice Pantaleo), relatore prof.M.Sozzi, che hanno studiato in dettaglio lo sviluppo e l'ottimizzazione dei metodi per la ricerca di cerchi singoli e multipli, e il vantaggio dell'impiego di un trigger basato su tali informazioni per la ricerca di eventi rari in NA62.

4.2 Misura e riduzione della latenza

Il tempo di calcolo non appare essere un grande problema, considerando che comunque scala con il numero di schede video impiegate: è sufficiente aumentare il numero di processori a disposizione per diminuire il tempo effettivo di calcolo per evento. In ogni caso per un trigger di livello 0 la latenza di calcolo è un parametro molto importante perchè da essa dipendono le caratteristiche hardware del readout. Ci sono due aspetti fondamentali nelle GPU che ne limitano le caratteristiche in termini di latenza. Il primo è legato al fatto che per ottenere buone prestazioni in termini di throuput di calcolo è necessario riempire il processore con "molto lavoro", ovvero saturare i core di calcolo con molti eventi contemporaneamente. Questo implica un certo tempo di gathering, ovvero tempo che è conveniente attendere prima di iniziare il calcolo in modo da sfruttare al meglio le caratteristiche del processore. Il secondo punto è il fatto che i dati vanno portati fisicamente sulla scheda grafica e questo ha i tempi tipici dettati dall'hardware utilizzato per la trasmissione. Ci sono vari punti in un PC standard in cui questo tempo viene speso. Per prima cosa la trasmissione dei dati attraverso i link di connessione (ad esempio ethernet nel caso di NA62), successivamente la copia dei dati dall'interfaccia di rete (NIC) fino alla memoria del PC attraverso il trasferimento sul bus PCI express, la copia dei dati dallo spazio di memoria della ricezione allo spazio di memoria utilizzabile dalle applicazioni e per ultima la copia dei dati, sempre attraverso il bus PCI express, dei dati dal PC alla scheda grafica (fig.4.3 e fig.4.4).

Nelle sotto sezioni successive si discuteranno i modi che stiamo studiando per diminuire questi tempi. Un aspetto importante delle misure effettuate è quello che non è facile,

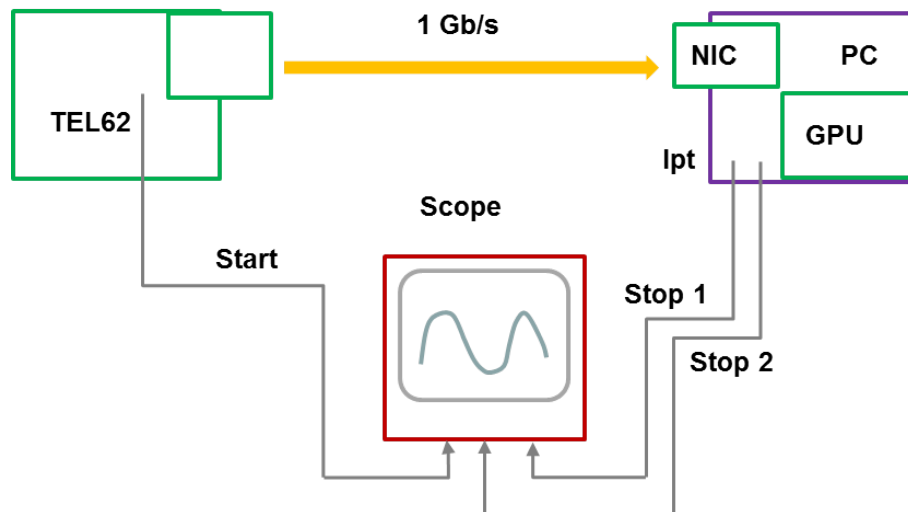


Figura 4.5: Schema della procedura di misura della latenza.

a priori, la misura delle componenti di latenza con risoluzione del microsecondo tra due sistemi, quello che produce i dati e quello che li elabora, che appartengono a domini di clock disgiunti. Per superare questa difficoltà abbiamo utilizzato uno schema come quello presentato in fig.4.5. I dati vengono prodotti da una scheda di readout dell'esperimento NA62 (la TEL62) e vengono elaborati da un PC con bus PCI express gen.2 e scheda TESLA K20. La misura della latenza viene effettuata con un oscilloscopio che riceve un segnale di start dalla TEL62 quando il pacchetto contenente i dati è inviato attraverso la connessione ethernet, e un segnale di stop dal PC, attraverso la porta parallela, quando l'elaborazione è terminata (oppure in vari step del processo di elaborazione).

4.2.1 PFRING

Come sopra spiegato la principale componente nella latenza totale viene dalla gestione delle copie effettuate dal driver standard della NIC. AL fine di diminuire la latenza dovuta a questa componente del trasporto dei dati, abbiamo sviluppato un framework multi-thread in cui sia possibile utilizzare un driver speciale per la cattura rapida dei pacchetti. Il framework disegnato consta di tre componenti

- *Network communication*
- *Job scheduler*
- *GPU kernel*

Il network communication è basato su *PF RING*[4]. *PF RING* è un framework per la cattura accelerata dei pacchetti che implementa l'allocazione di buffer memory-mapped nel momento della creazione del socket, al fine di copiare direttamente i pacchetti in arrivo. Per ottenere alte performance di cattura dei pacchetti in una simile struttura è necessario modificare i driver standard della NIC, in quanto tali driver non sono ottimizzati per la cattura. Un driver speciale, DNA (Direct NIC Access), è impiegato per gestire il ring buffer allocato al momento della creazione del socket. Tale ring buffer contiene i puntatori alla posizione del pacchetto nella memoria di ricezione. Per ottenere il contenuto del

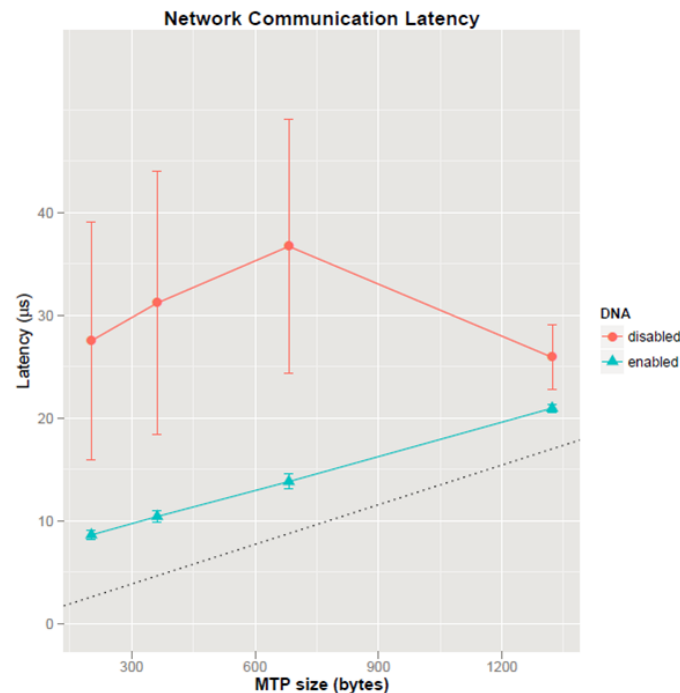


Figura 4.6: Confronto tra la latenza misurata con PFRING e con driver standard.

pacchetto, il trigger framework deve deferenziare il puntatore e aggiornarlo alla posizione del pacchetto successivo. I dati vengono direttamente copiati in user-space attraverso procedure DMA, e questo ottimizza le performance in termini di latenza. In fig.?? si mostra il confronto, in termini di latenza, tra l'utilizzo, con la stessa scheda di rete, di un driver normale e di un sistema basato su PFRING. Come si vede, oltre al miglioramento dovuto alla diminuzione del tempo impiegato per copiare i dati in user-space, si ha un netto miglioramento nella stabilità di tale latenza.

Il Job scheduler gestisce una coda intelligente strutturata in multi-producer e multi-consumer attraverso una struttura a threads. I producer sono i threads che gestiscono la comunicazione attraverso la gestione dei ring buffer di PFRING, mentre i threads consumer hanno il compito di preparare i dati per l'elaborazione sulla GPU. Tre threads consumer funzionano in modo concorrente su differenti set di dati per svolgere le seguenti funzioni

- I dati ricevuti dal un producer sono raccolti e preparati in ring buffer allocati nella pinned memory.
- quando un numero sufficiente di eventi (o un timeout) è raggiunto, il buffer di dati è inviato alla global memory della GPU e il processamento inizia.
- I risultati sono trasferiti all'host.

I tre compiti possono essere svolti contestualmente in quanto, nelle ultime versioni di GPU, questo è permesso dall'esistenza di strutture per la parallelizzazione del calcolo e della copia nelle due direzioni.

Il kernel GPU impiegato in questo tipo di struttura è assolutamente arbitrario, nel senso che il frame di invio dei dati descritto sopra, non dipende dalle caratteristiche del kernel

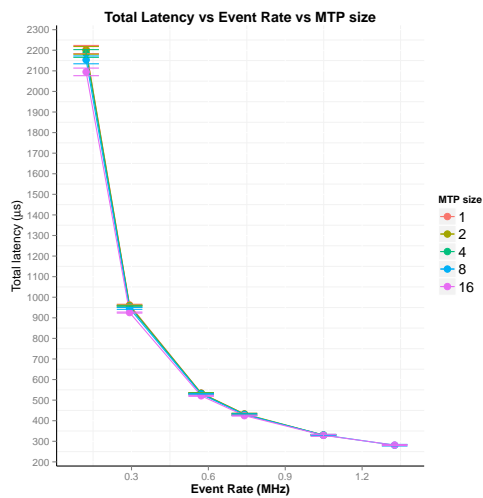


Figura 4.7: Latenza totale con la soluzione PFRING.

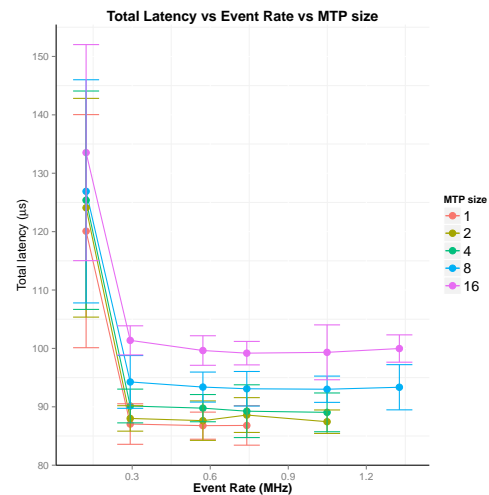


Figura 4.8: Componente della latenza escludendo il tempo di raccolta degli eventi (che dipende dal rate di produzione degli eventi stessi).

stesso. Ovviamente, però, le performance del sistema dipendono dal tempo di esecuzione del kernel in quanto l'esecuzione sulla GPU è non preemptive, per cui è sempre necessario aspettare il termine del calcolo.

Impiegando un kernel semplice, di tipo fit di singolo cerchio, abbiamo implementato un sistema completo con invio dei dati dalla TEL62 e calcolo in un PC fornito di TESLA K20 come sopra descritto. Nelle fig. ?? si possono vedere i risultati in termini di latenza ottenuta per pacchetti di 256 eventi in funzione del rate di produzione dei dati. Come si vede la latenza è fortemente dipendente dal tempo di gathering degli eventi, la componente dovuta al sistema è dell'ordine dei $100\mu s$ (come mostrato in fig.4.8).

4.2.2 NaNet

La scheda NaNet è basata sulla scheda APEnet+ sviluppata per permettere il trasferimento diretto dei dati dalla NIC alla memoria GPU senza buffering intermedio. Tale trasferimento avviene sfruttando una connessione peer-to-peer con la GPU e impiegando, quindi, in collegamento di tipo GPUDirect RDMA, che non necessita il passaggio dei dati nel processore del PC host. L'implementazione originaria è stata completata con l'aggiunta della parte di offloading del protocollo di trasmissione dei dati provenienti dalla TEL62 (UDP), al fine di gestire direttamente il controllo del flusso nel firmware della scheda (fig.4.9). NaNet è realizzata, per il momento, su un kit di sviluppo con Altera Stratix IV con connessione con il PC di tipo PCI express gen.2 . I dati vengono accumulati direttamente sulla memoria GPU e il kernel di calcolo, lo stesso impiegato per i test con la soluzione PFRING, viene eseguito appena il buffer raggiunge il numero di eventi predefinito. Le performance in termini di latenza di trasferimento dei dati sono state misurate in diversi modi, sia standalone al sistema stesso che nel sistema completo, con l'oscilloscopio esterno come sopra descritto. In particolare è stata misurata la banda totale del sistema (nella presente implementazione si usa un solo link a 1 Gb/s per il trasporto dei dati) come

4.2.3 Sistema “ibrido” per scopo di misura delle caratteristiche del PC

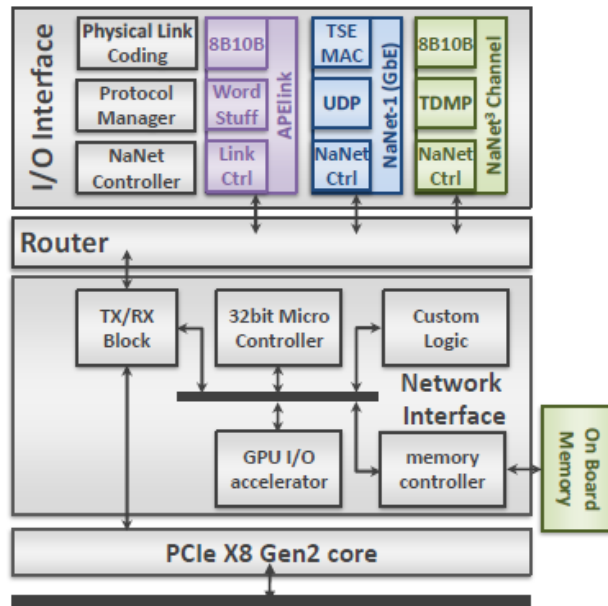


Figura 4.9: Schema a blocchi del funzionamento del Firmware di NaNet.

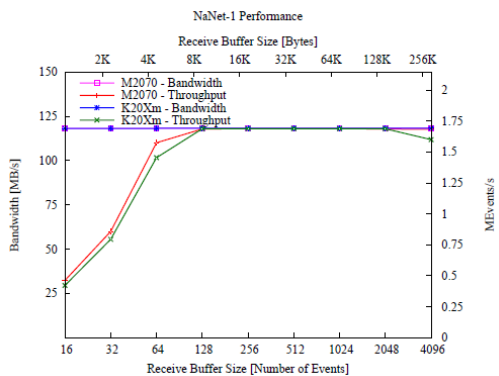


Figura 4.10: Banda di trasmissione ottenuta con la versione 1Gb/s della scheda NaNet.

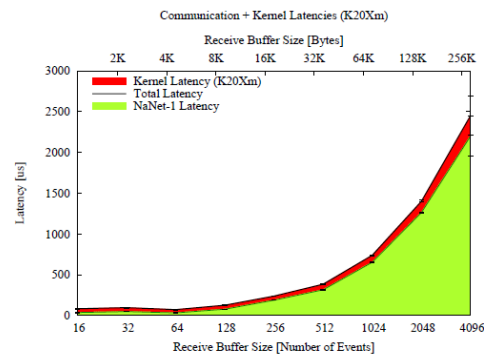


Figura 4.11: Latenza di trasmissione e di calcolo, misurata con la scheda NaNet-1.

mostrato in fig.4.10. In fig. 4.11 si mostra la misura della latenza ottenuta con la tecnica dell’oscilloscopio al variare del numero di eventi nel buffer di attesa. Anche in questo caso le performance sono tali da garantire il rispetto delle specifiche di NA62 (1 ms di latenza massima), per valori di buffering idonei al calcolo su GPU.

4.2.3 Sistema “ibrido” per scopo di misura delle caratteristiche del PC

Al fine di studiare le caratteristiche del bus di trasmissione, si è deciso di implementare anche un terzo sistema, di cui si occupa l’unità di Ferrara. Lo scopo di questo sviluppo

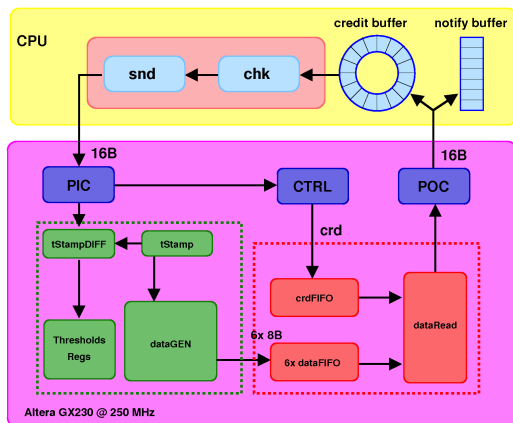


Figura 4.12: Schema del sistema utilizzato per misurare il tempo di copia dei dati attraverso il bus PCI express.

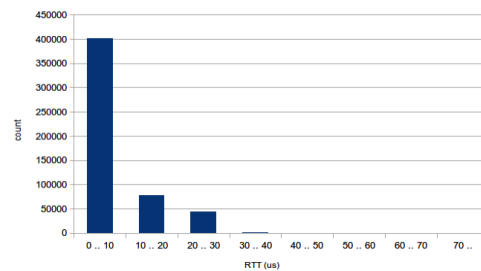


Figura 4.13: Risultati ottenuti per il Round Trip Time.

è quello di caratterizzare direttamente la velocità del bus PCI express in modo da avere un modo semplice per fare il benchmarking delle varie architetture che si basano sull'utilizzo dell'host per ricevere i dati (come nella soluzione PFRING o in una soluzione che non impiega nessun metodo di riduzione della latenza). L'unità di Ferrara nel corso di questi mesi ha lavorato con una scheda di sviluppo per FPGA dotata di Altera Stratix 4 e interfacce di rete gigabit ethernet, concentrandosi sul trasferimento dati tra la scheda e il computer di supporto. La scheda infatti è dotata di un connettore PCI-Ex x8 gen2. Per questa scheda è stato sviluppato un firmware che gira sulla fpga e il relativo driver linux che gira sul computer di supporto, per permettere alla scheda di assumere il controllo del bus PCI Ex e di trasferire dati direttamente nella ram del computer riservata dal driver (cioè il trasferimento viene iniziato dalla scheda, e non dal pc), segnalando al computer quali blocchi di memoria sono già pronti per essere analizzati. Al momento questo viene testato usando dei dati dummy generati dalla stessa fpga: stiamo lavorando per integrare le interfacce di rete a 1Gbps. In fig.4.12 è presentato uno schema della procedura utilizzata per effettuare la misura.

Le misure le stiamo effettuando programmando la fpga per aggiungere un proprio timestamp ai dati trasmessi al pc, e facendo in modo che il pc ritrasmetta le informazioni alla fpga che può così misurare il tempo utilizzato per compiere lo scambio di informazioni. Nel nostro schema partiamo anche dal presupposto che le informazioni che arrivano non siano necessariamente ordinate temporalmente, e che quindi la cpu del computer debba riallineare temporalmente tra loro i dati. Il tempo impiegato chiaramente dipende da quanto sono disallineati questi dati e da quali operazioni esattamente faccia la cpu del computer prima di rimandare i dati alla fpga. Nei nostri test abbiamo fatto in modo che il compito della cpu fosse il confrontare le eventuali correlazioni temporali con una lookup table di eventi di interesse e, come si può vedere in fig.4.13 nel caso peggiore riusciamo a impiegare meno di $50\mu s$.

Capitolo 5

GPU e HLT: stato

Gli studi rivolti all'impiego delle GPU in trigger di alto livello si innestano in una serie di studi già considerati in diversi esperimenti di HEP. Consci del fatto che negli upgrade di luminosità il rate di eventi utile crescerà notevolmente e sempre più sarà necessario di disporre di efficaci selezioni online, tutti gli esperimenti hanno iniziato a pensare a strategie per diminuire le dimensioni e i costi delle farm online. Altri esperimenti, come LHCb, hanno persino iniziato a riflettere sulla possibilità di selezionare direttamente gli eventi di interesse in PC con un ridottissimo livello hardware (con un architettura quasi triggerless).

Per quanto concerne gli sviluppi relativi alle applicazioni delle GPU nei livelli alti di trigger, per i primi 6 mesi del progetto abbiamo indicato i seguenti obiettivi:

- Identificazione di algoritmi di tracking in ATLAS e test su architettura GPU (Roma).
- Interfaccia con framework di ATLAS (Roma).
- Sviluppo dell'algoritmo di muon tracking su GPU e confronto con la versione attuale (Roma).
- Studio di pattern recognition per tracciatori al silicio (Ferrara).
- Studio dell'utilizzo delle GPU per il tracciatore nel trigger software di NA62.

Tali obiettivi sono stati parzialmente raggiunti per quanto riguarda lo studio dell'integrazione dell'elaborazione su GPU nel framework preesistente di elaborazione online di ATLAS. Nei prossimi mesi ci concentreremo a sviluppare più specificatamente gli algoritmi di utili per il trigger di muoni e a testarli rispetto alla versione attuale. Per quanto riguarda il trigger di NA62, alcuni studi preliminari sono stati fatti sul possibile impiego della trasformata di Hough in contesti multitraccia.

Rispetto agli impegni presi, abbiamo deciso di dedicare ulteriormente delle risorse alla collaborazione con il gruppo LHCb di Padova, per studiare l'impiego delle GPU nel trigger di alto livello di LHCb. Tale impegno rientra perfettamente negli scopi del progetto, ed è stato possibile grazie al fatto di poter disporre di risorse impreviste presso l'unità di Ferrara (il responsabile dell'unità è al momento pagato con soldi differenti da quelli del progetto come inizialmente previsto).

5.1 Studi preliminari per il trigger HLT di ATLAS

E' stato acquistato un server (gap01.roma1.infn.it), dotato di un processore multicore (CPU) Intel Xeon E5-2620 e di una scheda grafica (GPU) Nvidia GTX Titan.

Gli studi sulla parallelizzazione degli algoritmi di trigger di ATLAS sono rivolti principalmente al trigger di alto livello per la ricostruzione dei muoni. L'unità di Roma intende occuparsi principalmente dell'ottimizzazione di due algoritmi. Il primo seleziona un muone cercando di associare un segmento di traccia nello spettrometro per i muoni ad una traccia nel tracciatore interno (MU_COMB). Il secondo identifica il muone valutando il numero di tracce ricostruite in un cono di apertura data attorno alla sua direzione di volo (MU_ISO).

Questi algoritmi devono essere adattati alle condizioni di presa dati del secondo run di LHC, quando l'aumento di luminosità porterà ad un incremento del rate di tracce da ricostruire entro le stesse latenze. Per riuscire a mantenere le stesse efficienze di selezione di muoni a parità di eventi di fondo, è indispensabile rendere gli algoritmi di trigger più efficaci. E' necessario dunque aumentare la potenza di calcolo sia per continuare a fare l'associazione tra il segmento di traccia nel rivelatore a muoni e la traccia nello spettrometro interno in un regime con più alta molteplicità di tracce, che per calcolare l'isolamento del muone in coni di apertura maggiore.

L'unità di Roma si è impegnata verso la collaborazione ATLAS a portare avanti questi studi di fattibilità nell'ambito del gruppo di lavoro di ATLAS che si occupa del software di trigger dei muoni. In particolare, Matteo Bauce è impegnato in questa attività e lo sarà almeno fino alla primavera 2014. Da un lato questa attività consente a Matteo e al gruppo di inserirsi nei progetti di miglioramento del trigger di muoni per la futura presa dati, dall'altro, permette di acquisire delle competenze specifiche sugli algoritmi di ricostruzione di muoni. Quest'ultimo punto è fondamentale per individuare i limiti del software attuale e per sviluppare nuovi algoritmi intrinsecamente paralleli che risolvano questi limiti.

Come primo test delle performance della GPU si è misurata la latenza di esecuzione dell'algoritmo di fit delle tracce, che sfrutta delle candidate tracce grezze ricostruite da algoritmi di path finding e ne determina i parametri con maggiore risoluzione¹. L'algoritmo può essere ottimizzato eseguendo il fit di ciascuna traccia in thread paralleli; per massimizzare la potenza di calcolo della GPU le tracce da analizzare vengono raggruppate in *eventi*, e le tutte le tracce all'interno di un evento vengono processate in parallelo. Variando il numero di tracce per evento abbiamo confrontato il tempo necessario al processamento in modo sequenziale sulla CPU o in parallelo sulla GPU. In particolare, nelle figure 5.1 è riportato il tempo impiegato in media per ogni traccia (sopra) e per ogni evento (sotto) al variare del numero di tracce per evento. Si noti come nel grafico logaritmico (in alto a destra) sia apprezzabile la saturazione della capacità della GPU intorno a 5×10^3 tracce per evento. Questo valore è da confrontare con il numero di processori della GPU pari a $\sim 3 \times 10^3$.

Si osserva che mentre l'algoritmo sequenziale impiega lo stesso tempo per ciascuna traccia, il processamento su GPU beneficia del processamento in parallelo di un maggiore numero di tracce. Il tempo necessario a processare ciascun evento con la GPU è

¹Questo algoritmo è stato studiato ed adattato per l'esecuzione su GPU dal gruppo di ATLAS di Edimburgo.

5.1. Studi preliminari per il trigger HLT di ATLAS

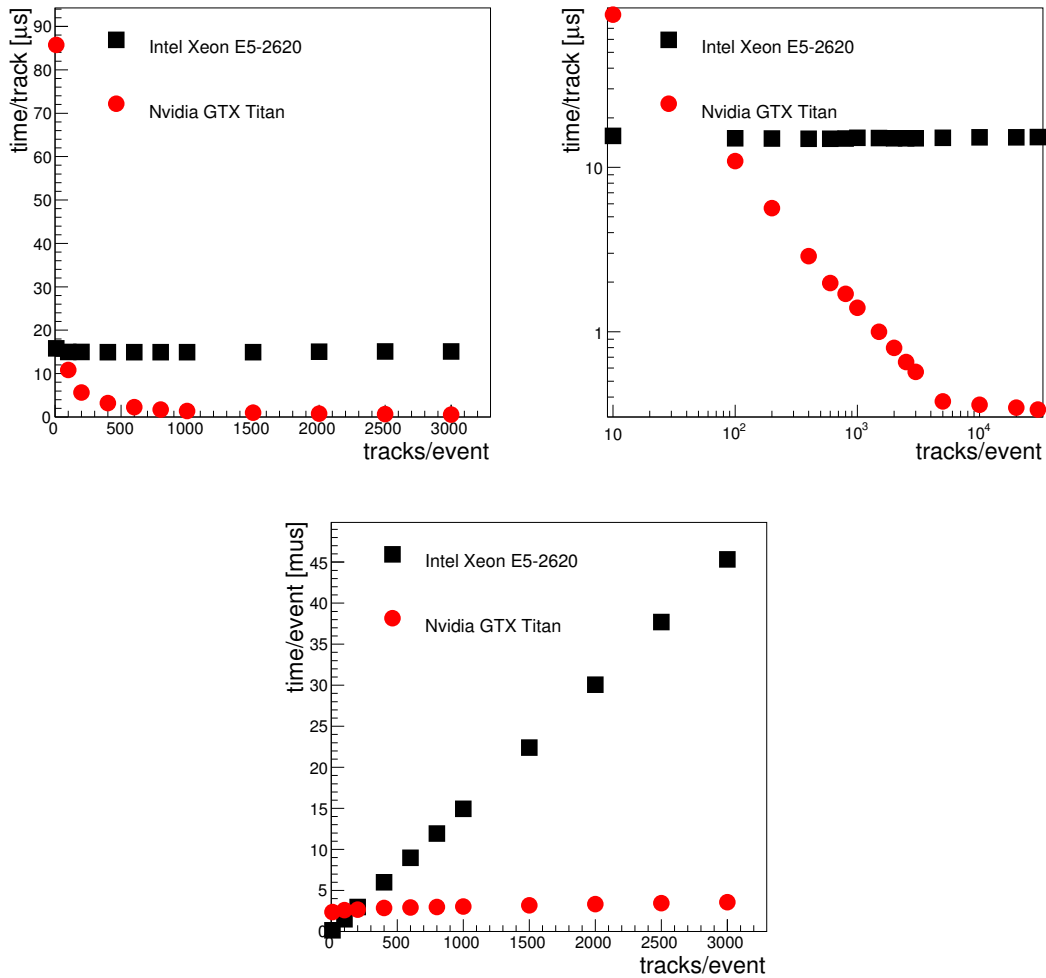


Figura 5.1: Latenza di esecuzione dell'algoritmo di fit eseguito sequenzialmente su CPU o in parallelo su GPU, in funzione del numero di tracce analizzate in ogni *evento*. Sopra, il tempo medio di processamento per traccia in scala lineare a sinistra e logaritmica a destra; sotto, il tempo medio di processamento per evento. Si noti come nel grafico logaritmico (in alto a destra) sia apprezzabile la saturazione della capacità della GPU intorno a 5×10^3 tracce per evento. Questo valore è da confrontare con il numero di processori della GPU pari a $\sim 3 \times 10^3$.

pressochè costante al variare del numero di tracce per evento, diventando vantaggioso rispetto all'algoritmo sequenziale, che cresce linearmente, quando si considerano centinaia di tracce per evento.

Da questi studi preliminari abbiamo potuto imparare come integrare il calcolo parallelo nel software di Atlas, isolando delle funzioni parallelizzabili che possono essere implementate per l'esecuzione su GPU, gestendo in modo proficuo le risorse di quest'ultima sia in termini di allocazione di memoria, che in termini di trasferimento di dati.

Nel prossimo futuro ci proponiamo di studiare le prestazioni di questi algoritmi di trigger dei muoni di ATLAS in modo analogo a quanto fatto finora per l'algoritmo di fit di traccia, includendo possibili miglioramenti dati dall'ottimizzazione dell'utilizzo della memoria della GPU ed il trasferimento dei dati dalla GPU alla CPU. In seguito sarebbe interessante studiare i miglioramenti che si possono ottenere implementando algoritmi più complessi per ricostruire online informazioni con precisione simile a quella raggiunta dal processamento offline.

5.2 Studi preliminari per il HLT trigger di LHCb

Si sta lavorando su due fronti, entrambi focalizzati sullo studio e applicazione di acceleratori grafici (GPU) per il trigger di alto livello di LHCb, in collaborazione con il gruppo di LHCb Padova. Il primo è il porting dell'algoritmo di ricostruzione delle tracce nel VELO, chiamato FastVelo. La traduzione dell'algoritmo in linguaggio "parallelo" sotto Cuda presenta qualche difficoltà legata al fatto che l'algoritmo seriale è stato implementato da una persona che attualmente non lavora più per LHCb e la documentazione è data dalla sola nota interna dello sviluppatore. Inoltre l'algoritmo non è mai stato pensato per essere eseguito in modo parallelo, quindi presenta caratteristiche intrinsecamente "seriali" che ne rendono difficile il porting. Il secondo fronte è lo studio e l'implementazione dell'algoritmo della trasformata di Hough per il track fitting. Anche se diversi studi hanno evidenziato che lo speedup risulta piuttosto basso rispetto ad altre tecniche di track fitting, la trasformata di Hough è un esercizio utile per affrontare i problemi legati al parallelismo degli algoritmi e delle strutture dati utilizzate. Inoltre ci si aspetta che in situazioni con elevato numero di tracce, l'algoritmo possa comunque risultare più efficiente di altri, senza contare che il meccanismo è lo stesso alla base dell'algoritmo retina di cui è stata proposta recentemente un'applicazione nell'LLT di LHCb e che dovrebbe rappresentare la naturale prosecuzione di questo studio.

5.3 Studi preliminari per il trigger di NA62

Come studio preliminare per il trigger HLT di NA62 si sta studiando la possibile applicazione delle GPU nella ricostruzione online dello spettrometro magnetico. Tale spettrometro ha delle caratteristiche specifiche che rendono l'applicazione di algoritmi standard di ricerca di tracce non banale. Abbiamo implementato procedimenti basati sulla trasformata di Hough (fig.5.2). La risoluzione ottenuta con questa procedura è compatibile con quella della ricostruzione offline (fig.5.3) anche per eventi con molte tracce, ma l'efficienza, a causa in particolare di eventi con sciame e a particolari configurazioni geom-

5.3. Studi preliminari per il trigger di NA62

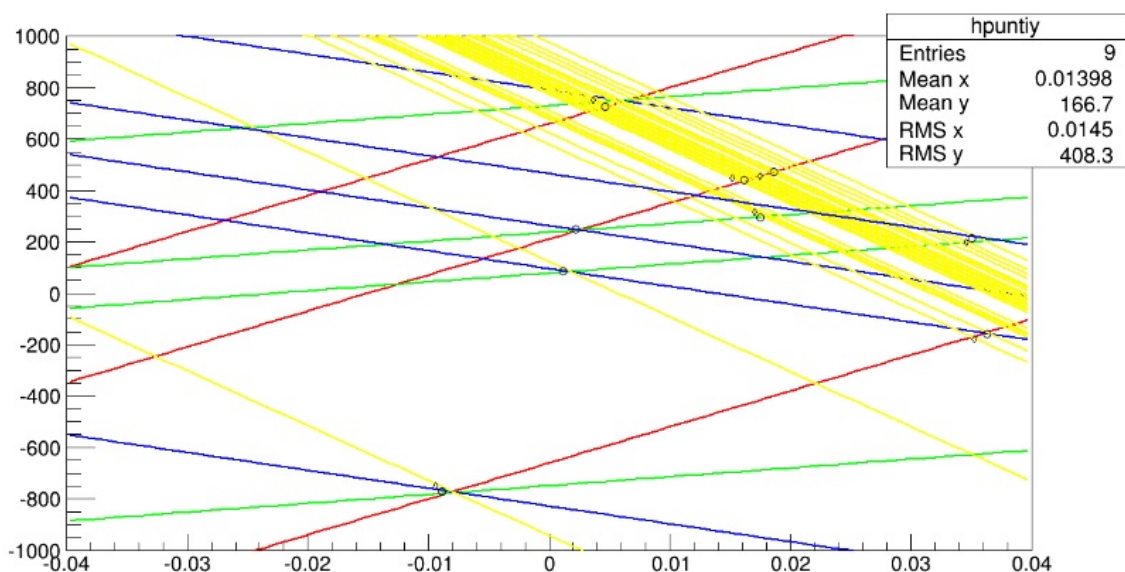


Figura 5.2: Esempio di 3 tracce identificate sullo spettrometro. Ogni colore rappresenta una diversa camera. Sugli assi ci sono p e q , la pendenza e l'intercetta delle tracce nello spazio ordinario.

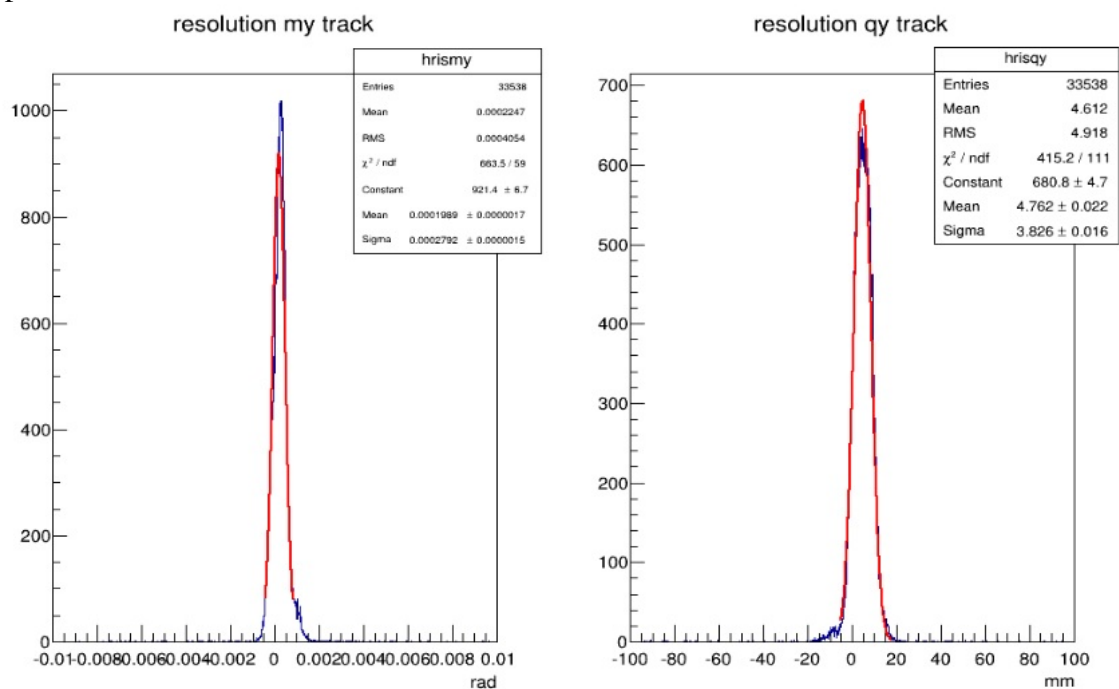


Figura 5.3: La risoluzione sulla pendenza e sull'intercetta delle tracce ricostruite è compatibile con quella della ricostruzione offline

triche in cui una o più tracce sono identificabili con un numero minori di hit, deve essere sostanzialmente migliorata.

Capitolo 6

Ricostruzione immagini mediche

La ricostruzione online delle immagini mediche utilizzando i processori video in varie metodologie di acquisizione, è il secondo importante task del progetto. Questo obiettivo vede coinvolte le diverse unità di ricerca, con argomenti per molti versi complementari ma che hanno, potenzialmente, alcuni punti in comune che verranno, eventualmente, considerati durante lo svolgimento del progetto.

Per quanto riguarda la ricostruzione delle immagini mediche, abbiamo identificato e indicato ai revisori, per i primi sei mesi, i seguenti obiettivi principali:

- Confronto tra PET e CT per identificare algoritmi comuni idonei alla parallelizzazione su GPU (Ferrara+Pisa).
- Versione prototipale di algoritmo per PET (Pisa).
- Realizzazione delle mappe di diffusione per NMR (Roma).
- Traduzione di algoritmi paralleli NMR da versione matlab a versione C (Roma).
- Implementazione di proiettori e retroproiettori per CT su GPU (Ferrara).
- Implementazione algoritmo FDK su GPU (Ferrara).

Per quanto riguarda NMR e CT gli obiettivi sono parzialmente raggiunti, nel senso che molto lavoro preliminare di sviluppo dei prototipi è stato fatto e, al momento, sono in fase di definizione i codici completi su alcuni degli algoritmi proposti. Per quanto riguarda la PET ci sono stati degli studi di fattibilità e paragone con algoritmi impiegabili nella CT.

6.1 Risonanza magnetica nucleare

Due desk collegati al server GAP01 sono stati recentemente acquistati ed installati nel laboratorio NMR. Attualmente si sta sviluppando il software necessario per processare e ricostruire le mappe cerebrali in diffusione non-Gaussiana. Tuttavia abbiamo realizzato dei tests preliminari per stimare il guadagno in termini di tempo di calcolo per il processamento delle immagini. Abbiamo simulato il segnale NMR in ogni pixel di immagini a diversa risoluzione (ovvero matrici di diversa dimensione) da matrici 32x32 a

matrici 2048×2048 . Le immagini rappresentano una sezione di 1mm^2 di sostanza bianca cerebrale contenente fasci coerenti di assoni dell'ordine di 2×10^3 assoni. Risolvendo le equazioni di Bloch Torrey tenendo conto della presenza di disomogeneità di campo magnetico indotte all'interfaccia fra assoni e acqua diffondente fra gli stessi assoni, è stato ricostruito numericamente l'andamento del segnale NMR al variare del parametro b che "pesa" il segnale in diffusione. Sono stati utilizzati trenta valori di b da 0 a 5000 s/mm^2 . Il segnale è stato analizzato secondo il modello di diffusione non Gaussiana basato sulla Kurtosis diffusionale. Il grafico in figura 6.1 riporta il tempo necessario per l'analisi completa in funzione della dimensione della matrice. È stato realizzato utilizzando un algoritmo opportunamente parallelizzato per essere eseguito su CPU multithread (8 thread) e GPU con supporto CUDA, sfruttando il Parallel Computing Toolbox di MATLAB (versione R 2012 b). Le dimensioni tipiche di immagini mediche NMR sono in genere al massimo di 512×512 pixels. Dal grafico, il guadagno atteso in termini di tempo di calcolo varia da circa 102 per immagini in bassa risoluzione fino ad arrivare a 104 per immagini di 512×512 pixels.

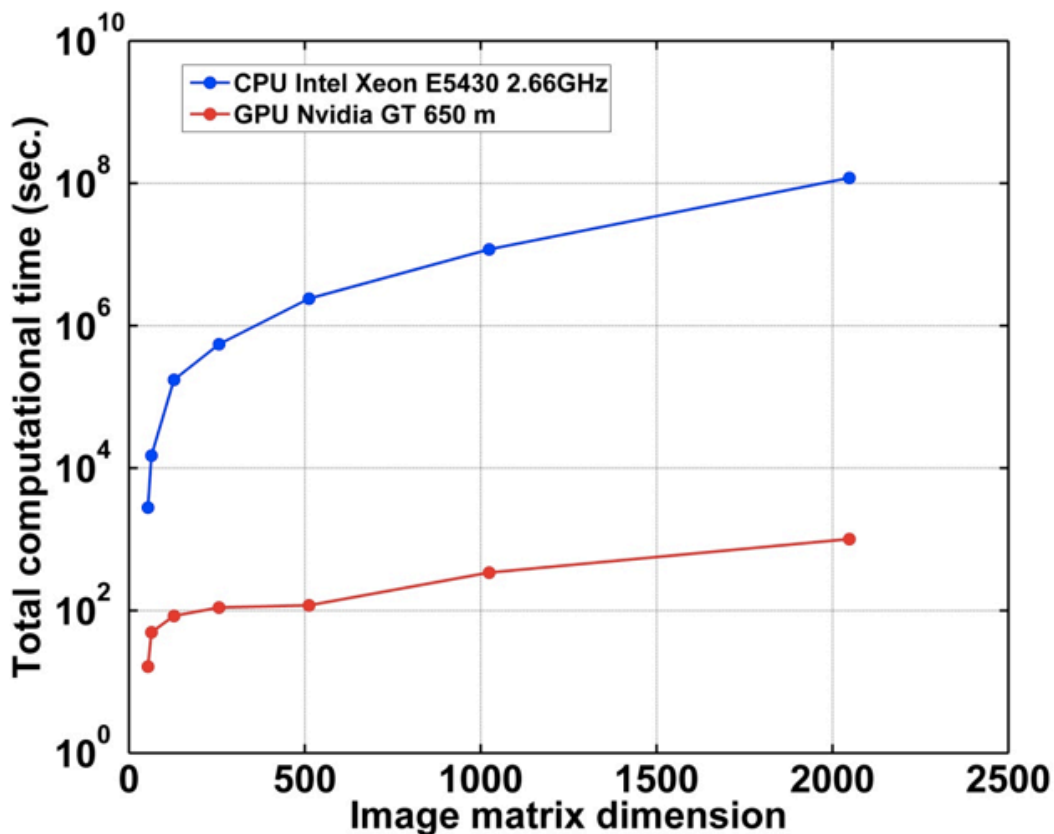


Figura 6.1: Tempo necessario per l'analisi completa in funzione della dimensione della matrice.

Poiché nella reale ricostruzione delle immagini occorre impiegare in maniera iterativa (dalle 40 alle 400 iterazioni) algoritmi di minimizzazione di funzioni non lineari, il reale guadagno in termini di tempo di elaborazione è stimato essere ridotto di un fattore variabile da 40 a 400. Sulla base di questo test preliminare ci aspettiamo un guadagno reale

per l'elaborazione delle immagini NMR in diffusione non-Gaussiana compreso fra 20 e 102 per immagini 128×128 .

6.2 Tomografia assiale (CT)

All'interno del progetto FIRB il gruppo di Fisica medica dell'Università di Ferrara è impegnata nello sviluppo di algoritmi per la ricostruzione di dati provenienti da sistemi CT. In particolare siamo interessati alla ricostruzione di dati provenienti da tomografi che utilizzano la geometria di tipo Cone Beam (CBCT). Durante la prima parte del progetto FIRB il lavoro di ricerca si è indirizzato sui temi discussi nelle sottosezioni.

6.2.1 Sviluppo di un codice per le trasformazioni di proiezione e retro-proiezione per applicazioni tomografiche implementata su processori grafici NVIDIA.

Nel nostro caso l'operatore di proiezione deve calcolare la seguente trasformazione

$$g(u, v, \alpha) = P_\alpha[f(x, y, z)]$$

dove la funzione $f(x, y, z)$ rappresenta il coefficiente di attenuazione del campione nel punto di coordinate (x, y, z) , mentre la funzione $g(u, v, \alpha)$ rappresenta l'integrale lungo la linea di proiezione individuata dalla sorgente e dal punto (u, v) nel piano del rivelatore posizionato ad un angolo α . L'operatore di retro-proiezione è l'operatore aggiunto di P_α^\dagger che permette di calcolare $f_\dagger(x, y, z)$ a partire da $g(u, v, \alpha)$. Esistono diversi modelli di proiettori: noi abbiamo scelto di implementare un proiettore di tipo Siddon [5]. Al momento, è stata completata la versione 0.1 del codice funzionante su GPU per entrambe le trasformazioni. Per la trasformazione di proiezione, il codice è stato organizzato in modo da assegnare ad ogni thread CUDA il compito di calcolare l'integrale lungo una linea di proiezione individuata da (u, v, α) . Nel caso della trasformazione di retro-proiezione ad ogni thread Cuda è assegnato il compito di determinare il contributo di un singolo volume di ricostruzione individuato dalle coordinate (x, y, z) . Riguardo l'ottimizzazione della gestione della memoria presente sulla scheda grafica si è scelto di allocare, quando possibile, tutto il volume di ricostruzione o una sua partizione sul device grafico e caricare invece un numero limitato di proiezioni, tipicamente 2 o 3, che vengono elaborate prima di passare alle proiezioni successive. Nel caso di dati floating point, con una scheda grafica dotata di 4 GB di memoria, si possono elaborare direttamente volumi fino a 512^3 , mentre per i volumi delle dimensioni maggiori di 1024^3 bisogna ricorrere alla partizione del volume. I primi test hanno confermato un guadagno di un fattore superiore a 50 nei tempi di proiezione se confrontati con un singolo thread di una CPU.

6.2.2 Implementazione dell'algoritmo di FDK per la ricostruzione di dati di CBCT.

La ricostruzione dei dati provenienti da CBCT è al momento eseguita utilizzando l'algoritmo approssimato di Feldkamp-Davis-Kress (FDK) [6] che permette una ricostruzione

diretta (non iterativa) dei dati. Le operazioni necessarie alla sua implementazione prevedono una fase di filtraggio 1D ed una di retroproiezione. Il lavoro di sviluppo è in fase iniziale ed è orientato all'ottimizzazione dell'operazione di convoluzione necessaria per l'applicazione dei filtri di ricostruzione alle proiezioni acquisite. In questo caso, si è deciso di assegnare ad ogni thread CUDA il calcolo di un singolo elemento del vettore al quale applichiamo la convoluzione. Comunque è previsto anche un confronto sulle prestazioni utilizzando la libreria CUFFT di CUDA. Per l'operazione di retroproiezione è possibile utilizzare il codice sviluppato nel punto precedente.

6.3 Tomografia a Positroni (PET)

Per la ricostruzione delle immagini PET a Pisa è stato scelto di implementare l'algoritmo MLEM (Maximum Likelihood Expectation Maximization) che è uno dei più comuni per quanto riguarda la ricostruzione PET e che consente di ricostruire immagini allo stato dell'arte. Confrontando i punti in comune tra la ricostruzione PET e quella CT che verrà implementata dall'unità di Ferrara, è stato inoltre stabilito di implementare in una prima fase un framework comune per la ricostruzione di immagini che implementi le operazioni di retroproiezione e proiezione che sono comuni a tutti i tipi di ricostruzione. In futuro si dovrà valutare se questa astrazione comporti un'eccessiva perdita di efficienza computazionale ed eventualmente scegliere di implementare delle soluzioni dedicate.

Capitolo 7

Conferenze e Scuole

Abbiamo ritenuto utile, seppure in fase iniziale del progetto, partecipare ad alcune conferenze di Fisica e di strumentazione, al fine di presentare il lavoro finora svolto, soprattutto nell'ambito dei nostri studi sull'impiego delle GPU nel L0, e indicare alla comunità il nostro interesse a partecipare agli sviluppi che provengono anche da altri gruppi in molteplici esperimenti per studiare l'impiego delle GPU nei livelli alti di trigger. Molto brevemente riportiamo le conferenze a cui abbiamo partecipato, con qualche dettaglio sui talk o poster presentati e le motivazioni.

Workshop on GPU in High Energy Physics - Amburgo - 15-16 Aprile 2013

Questo workshop [7], organizzato a Desy, è stato fortemente centrato sulle tematiche riguardanti il nostro progetto. Hanno partecipato al workshop i tre responsabili delle unità (Massimiliano Fiorini, Gianluca Lamanna e Andrea Messina) con un talk di Gianluca Lamanna dal titolo "The NA62 Low Level GPU trigger" [8] vantaggi dell'uso delle GPU nel trigger di basso livello dell'esperimento NA62 e i risultati preliminari ottenuti sugli algoritmi utilizzabili per il contatore RICH.

ACAT 2013 - Pechino - 16-21 Aprile 2013

ACAT è una conferenza appartenente ad una serie iniziata nel 1990, per discutere di "Advance Computing and Analysis Technique in Physics" [9]. Il nostro talk dal titolo "GPU for Real Time processing in HEP trigger systems" [10] è stato dato da Piero Vicini (Gruppo Apenet - Roma), con lo scopo di presentare il progetto GAP e le attività ad esso correlate. Per questa conferenza sono stati scritti i proceedings.

EPS-HEP 2013 - Stoccolma 18-24 Luglio

EPS-HEP [11] è una delle più importanti conferenze di fisica delle particelle. Tra le sessioni parallele è prevista una sessione dedicata ai detector e in particolare un giorno è stato dedicato a talk riguardanti i sistemi di trigger con particolare riguardo agli upgrade e alle idee per sistemi futuri. Gianluca Lamanna ha presentato in un talk dal titolo "GPU for Real Time processing in HEP trigger systems" [12] i risultati ottenuti sul controllo

della latenza utilizzando il driver ottimizzato per la cattura dei pacchetti discusso in 4.2.1. Per questa conferenza sono stati scritti i proceedings.

CHEP 2013 - Amsterdam 14-18 Ottobre

Questa conferenza [13], giunta alla 20esima edizione, è dedicata al computing in fisica delle alte energie. Abbiamo avuto modo di presentare il nostro lavoro in un poster dal titolo “GPU for Real Time processing in HEP trigger systems” presentato da Roberto Ammendola (Gruppo Apenet) e un talk dal titolo “NaNet: a low-latency NIC enabling GPU-based, real-time low level trigger systems” presentato da Alessandro Lonardo (Gruppo Apenet). Il poster ha mostrato aspetti generali dell’impiego delle GPU in fisica delle alte energie, mentre nel talk si è discusso, in particolare, aspetti legati all’impiego della soluzione basata sul trasferimento dei dati peer-to-peer come discusso nella sezione 4.2.2. Per questa conferenza sono stati scritti i proceedings.

TWEPP - Perugia 23-27 Settembre

Questo workshop annuale [14] è dedicato a vari aspetti connessi all’elettronica in esperimenti di fisica delle alte energie. Abbiamo avuto la possibilità di presentare un poster centrato soprattutto sugli aspetti più connessi all’hardware nel nostro progetto, ovvero alla scheda Nanet, dal titolo “NaNet: a flexible and configurable low-latency NIC for real-time trigger systems based on GPUs”. Il poster è stato presentato da Andrea Biagioni (Gruppo Apenet). Per questa conferenza sono stati scritti i proceedings.

IEEE-NSS - Seoul 27/10-3/11 2013

La conferenza IEEE-NSS [15] è la conferenza di riferimento per l’elettronica in ambito di fisica delle particelle. Massimiliano Fiorini ha presentato un poster dal titolo “GPU for Real Time processing in Particle Physics” mostrando il confronto tra le due tecniche di riduzione della latenza presa in considerazione. Per questa conferenza sono stati scritti i proceedings.

Partecipazione alla Scuola di Bertinoro

La scuola di calcolo scientifico dell’INFN di Bertinoro ESC2013 [16] è stata dedicata, in buona parte, al calcolo parallelo e in particolare al calcolo sulle GPU. Abbiamo ritenuto utile inviare, come studenti, alcuni giovani collaboratori del progetto GAP e in particolare: Jacopo Pinzino, Cristiano Santoni e Matteo Bauce.

Capitolo 8

Problemi, difficoltà e proposte

Ci pare giusto, nello scopo di questo report, riportare le principali difficoltà che abbiamo avuto fino ad ora e che vediamo come possibili nel breve termine:

1. Vogliamo sottolineare che l'eventuale possibile spostamento del Principal Investigator dall'unità di Pisa ad altra sede (come si sta discutendo in questi giorni) non avrebbe, ovviamente, impatto nullo sul progetto. Il progetto è stato strutturato in modo da poter coinvolgere ricercatori già interessati e presenti nelle sezioni e dipartimenti inclusi. In particolare a Pisa, in cui molti degli obiettivi del programma proposta implicano attività di laboratorio, il contributo del PI, non solo come coordinatore, rimane essenziale. Come essenziale rimane la necessità che il PI interagisca frequentemente con i collaboratori dell'unità. La soluzione del problema, dal lato GAP, sarà discussa nell'immediato futuro, ma sicuramente implicherà uno sforzo aggiuntivo in termini del budget del progetto (prendere più personale non previsto nell'unità di Pisa, prendere personale per affiancare il PI nell'eventuale nuova sede dove verrà trasferito, trasferite del PI, etc.) e di efficacia nel lavoro e nel coordinamento. Ci aspettiamo che l'INFN contribuisca ad aiutarci a risolvere questo problema e chiediamo alla CSN1 di farsi interprete di questa nostra difficoltà futura.
2. La situazione che riguarda lo sviluppo della ricostruzione delle immagini per la PET a Pisa è, al momento, non adeguatamente supportata. Nonostante gli impegni presi, alcuni cambiamenti avvenuti tra la fase di presentazione e approvazione del progetto hanno fatto in modo che le risorse umane preventivate non siano adeguatamente disponibili. Nel futuro si punterà a coinvolgere altre persone e, possibilmente, ad assegnare risorse per impiegare personale in questo settore. Questo tipo di budget non era comunque previsto nel progetto per cui ci aspettiamo di provare a coinvolgere gruppi interessati ad un possibile cofinanziamento.
3. Nel progetto abbiamo citato una collaborazione con il policlinico Umberto I di Roma. L'approccio metodologico alla ricerca in campo medico è differente da quello che strettamente ci riguarda. I collaboratori afferenti a questa struttura hanno confermato la loro disponibilità, ma più nella fase di test che nella fase di sviluppo per gli algoritmi NMR. Inoltre i campi di studio dell'unità NMR di Roma e dell'ospedale sono differenti (nel primo caso NMR dell'encefalo, nel secondo total-body),

per cui punteremo a cercare punti di convergenza, che però, al momento, non sono stati discussi.

4. Per il momento abbiamo avuto un successo relativamente basso nel cercare di coinvolgere persone afferenti a settori diversi (informatici e elettronici). Vogliamo segnalare più una difficoltà “generica” che un problema, sapendo benissimo che questa cosa dipende fortemente dai progetti specifici e dai contatti pregressi. Sempre nello spirito di queste note, vogliamo solo sottolineare il fatto che non sarebbe male, in generale, se l’INFN trovasse il modo di “parlare” con settori differenti della ricerca (che vanno dalla fisica dei materiali, alla chimica, all’informatica, etc.) immaginando delle strutture e degli schemi in cui accordi singoli possano essere inquadrati.
5. Ci piacerebbe cercare di instaurare una collaborazione scientifica anche con altri gruppi interessati agli sviluppi che proponiamo. Siamo convinti che il nostro lavoro possa interessare molti degli sviluppi e dei rivelatori futuri, in quanto si inserisce in un settore (il calcolo parallelo) che, a qualche livello che sia trigger o ricostruzione o analisi, dovrà essere considerato per qualunque tipo di applicazione. Al momento questa collaborazione è molto forte con il gruppo APENet+ di Roma e ci auguriamo di iniziare a discutere in modo più approfondito anche con il gruppo LHCb di Padova. Auspichiamo che si possa tentare (in realtà “ritentare”) un coordinamento e una sinergia tra attività di questo tipo, all’interno della CCR oppure in un progetto di coordinamento specifico che coinvolga non solo le attività di CSN5 ma anche direttamente la CSN1.
6. Riteniamo che il sito del progetto possa essere un valido strumento per la coordinazione e la condivisione delle informazioni e delle attività nelle varie unità, sia per ragioni amministrative che scientifiche. Da non trascurare poi l’aspetto di presentazione verso l’esterno, per far conoscere la nostra attività e i risultati raggiunti. Al momento però, a causa di mancanza di tempo per dedicarci allo sviluppo dello stesso, il sito risulta fortemente incompleto oltre il fatto che è solamente in Italiano. Abbiamo in programma di ampliare la parte informativa e di sviluppare una parte “riservata” ai collaboratori in modo da parte condividere informazioni e note interne.

Capitolo 9

Futuro e Conclusioni

In questa relazione abbiamo presentato i risultati ottenuti dopo 8 mesi di attività inquadrando negli scopi generali del progetto.

Gli obiettivi proposti e indicati ai referee INFN sono stati parzialmente raggiunti: in particolare sono stati implementati due metodi per il trasporto dei dati a bassa latenza per quanto riguarda il L0 e ne sono state misurate le performance, è iniziato lo studio per l'integrazione di codice per GPU nel framework di ATLAS, è iniziato lo studio della possibile integrazione in LHCb, sono stati sviluppati codici prototipali per la ricostruzione delle immagini in NMR e CT. Abbiamo anche illustrato alcune delle ragioni che hanno generato alcuni ritardi che speriamo di recuperare a breve tempo. Insieme a questo abbiamo indicato i possibili sviluppi del prossimo periodo in accordo con la pianificazione globale del progetto.

Nella prossima riunione nazionale GAP, a Gennaio a Pisa, avremo modo di discutere insieme a tutti i collaboratori, la strategia e gli obiettivi dei prossimi mesi, e avremo modo di riportare le decisioni prese alla prossima riunione della commissione 1 all'inizio di Febbraio. In ogni caso possiamo anticipare che le cose più rilevanti che affronteremo nel prossimo periodo riguarderanno il confronto tra i risultati ottenuti per quanto riguarda la riduzione delle latenza a L0 e l'implementazione di algoritmi prototipali per quanto riguarda l'HLT. Si inizieranno a disegnare dei sistemi "dimostratori" per provare le nostre soluzioni "sul campo". Nell'ambito della ricostruzione di immagini punteremo a continuare il lavoro di sviluppo di algoritmi e inizieremo i primi test su immagini acquisite. Per proseguire in questo campo saranno necessari alcuni acquisti, in particolare di licenze software e altro che dettaglieremo meglio nel seguito.

Dal punto di vista del personale, all'inizio del prossimo anno, verrà acquisito, dall'unità di Pisa, un'ulteriore unità di personale. Questo punto ancora non è molto chiaro perchè dipende molto dalla logistica legata ad un possibile trasferimento di sede del PI. Questo fatto, comunque, come sottolineato nel presente documento, non avrà un effetto nullo sul progetto. Cercheremo di fare in modo di ridurre l'impatto di questa cosa, ma ci aspettiamo possibili aiuti e suggerimenti da parte dell'INFN, così come ci aspettiamo indicazioni e suggerimenti sulle altre difficoltà discusse.

Bibliografia

- [1] G. Collazuol, G. Lamanna, J. Pinzino and M. S. Sozzi, Nucl. Instrum. Meth. A **662** (2012) 49.
- [2] <http://www.tomshardware.com/t/gpus/>
- [3] B. Angelucci, G. Anzivino, C. Avanzini, C. Biino, A. Bizzeti, F. Bucci, A. Cassese and P. Cenci *et al.*, Nucl. Instrum. Meth. A **621** (2010) 205.
- [4] L. Deri, Improving Passive Packet Capture: Beyond Device Polling, Proceedings of SANE 2004
- [5] “Fast calculation of the exact radiological path for a three-dimensional CT array”, Robert L. Siddon , Med. Phys. 12, 252 (1985).
- [6] “Practical cone-beam algorithm”, L. A. Feldkamp, L. C. Davis, and J. W. Kress, JOSA A, Vol. 1, Issue 6, pp. 612-619 (1984).
- [7] <https://indico.desy.de/conferenceTimeTable.py?confId=7143#20130415>)
- [8] <https://indico.desy.de/materialDisplay.py?contribId=4&sessionId=2&materialId=slides&confId=7143>
- [9] <http://acat2013.ihep.ac.cn/>
- [10] <http://indico.ihep.ac.cn/getFile.py/access?contribId=46&sessionId=3&resId=0&materialId=slides&confId=2813>
- [11] <http://eps-hep2013.eu/>
- [12] <http://indico.cern.ch/getFile.py/access?contribId=187&sessionId=20&resId=0&materialId=slides&confId=218030>
- [13] <http://www.chep2013.org/>
- [14] <http://indico.cern.ch/conferenceDisplay.py?confId=228972>
- [15] <http://www.nss-mic.org/2013/NSSMain.asp>
- [16] <https://web2.infn.it/esc13/>