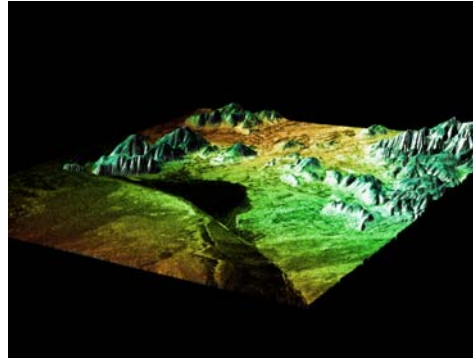
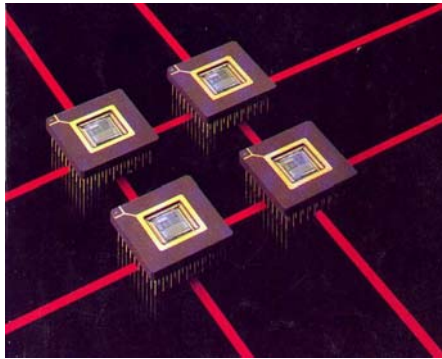


Attività:

Piattaforma distribuita ad alte prestazioni

Questa attività nasce dall'unità di intenti di CNR e ASI nel settore dei sistemi e delle applicazioni ad alte prestazioni.

L'obiettivo generale è quello di unire, alle caratteristiche di una architettura a componenti, le caratteristiche di architetture hardware-software ad alte prestazioni. Questo obiettivo si esplicherà nella realizzazione di un *prototipo* innovativo, reso disponibile alla comunità scientifica per future ricerche, oltre che alla comunità industriale ed a varie classi di utenza. L'attività ha caratteristiche verticali: il prototipo di piattaforma, che "si appoggia" su componenti standard (hardware-software-middleware) di ampia diffusione ed alta qualità, si estende su *tutti i livelli* (supporti architetture e a tempo di esecuzione, strumenti e ambienti di programmazione, componenti specializzati per librerie e software applicativo) integrando tali componenti standard secondo un approccio che mira contemporaneamente alla generalità e portabilità ed alle prestazioni. Inoltre, fondamentale per questa attività è lo sviluppo di una *applicazione*, concepita per la dimostrazione e la sperimentazione, scelta nel settore del *controllo ambientale e protezione civile*.



1.	Struttura del progetto ed enti partecipanti.....	2
2.	Obiettivo	4
	2.1. Contesto scientifico e tecnologico	4
	2.2. Approccio e metodologia.....	4
	2.3. Base di partenza.....	5
3.	Tematiche e strutturazione del progetto in task.....	6
4.	Task 1: Ambiente di programmazione portabile a componenti parallele.....	7
	4.1. Obiettivo generale.....	7
	4.2. Livello L1: supporti ad alte prestazioni	8
	4.3. Livello L2: ambiente e strumenti	8
	4.4. Base di partenza.....	9
5.	Task 2: Componenti ad alte prestazioni per Data/Web Mining e motori di ricerca	9
	5.1. Ambiente per Knowledge Discovery Process	10
	5.2. Ambiente per Question Answering	11
	5.3. Motori di Ricerca per dati semi-strutturati	11
6.	Task 3: Librerie scientifiche.....	12
	6.1. Contesto scientifico e tecnologico	12
	6.2. Obiettivi e metodologia	12
	6.3. Punto di partenza	13
7.	Task 4: Librerie per l'elaborazione di immagini	14
	7.1. Contesto scientifico e tecnologico	14
	7.2. Obiettivo e metodologia.....	14
	7.3. Punto di partenza	15
	7.4. Funzionalità e strumenti.....	15
8.	Task 5: Applicazione in ambito di controllo ambientale e protezione civile.....	16
	8.1. Integrazione di librerie per elaborazione di immagini telerilevate	16
	8.2. Catena di processamento SAR interferometrico	16
9.	Indicatori per la valutazione del raggiungimento degli obiettivi di progetto.....	17
10.	Articolazione temporale del progetto	17
11.	Riferimenti.....	18

1. Struttura del progetto ed enti partecipanti

Responsabile: Marco Vanneschi, Dipartimento di Informatica, Università di Pisa

Task 1: Ambiente di programmazione portabile a componenti parallele

Coordinatore: Marco Danelutto, Dipartimento di Informatica, Università di Pisa

Task 2: Componenti ad alte prestazioni per Data/Web Mining e motori di ricerca

Coordinatore: Franco Turini, Dipartimento di Informatica, Università di Pisa

Task 3: Librerie scientifiche

Coordinatore: Almerico Murli, CPS-CNR, Napoli

Task 4: Librerie per l'elaborazione di immagini

Coordinatore: Andrea Clematis, IMA-CNR, Genova

Task 5: Applicazione in ambito di controllo ambientale e protezione civile

Coordinatore: Giovanni Milillo, ASI, Matera

Task 6: Documentazione e divulgazione dei risultati del progetto

Coordinatore: Marco Vanneschi, Dipartimento di Informatica, Università di Pisa

Task 1: Ambiente di programmazione portabile a componenti parallele

Coordinatore: Marco Danelutto, Dipartimento di Informatica, Università di Pisa

Partecipanti:

Dipartimento di Informatica, Università di Pisa (DI UniPi)

Responsabile: Marco Danelutto

IFCAI, CNR, Palermo

Responsabile: Alberto Machì

Dipartimento di Informatica, Università di Venezia Cà Foscari (DI UniVe)

Responsabile: Salvatore Orlando

ISTI, CNR, Pisa

Responsabile: Domenico Laforenza (CNUCE)

Dipartimento di Elettronica e Informazione, Politecnico di Milano (DEI PoliMi)

Responsabile: Giuseppe Serazzi

Synapsis srl, Livorno

Responsabile: Riccardo Fontanelli

Task 2: Componenti ad alte prestazioni per Data/Web Mining e motori di ricerca

Coordinatore: Franco Turini, Dipartimento di Informatica, Università di Pisa

Partecipanti:

Dipartimento di Informatica, Università di Pisa (DI UniPi)

Responsabile: Franco Turini

ISTI, CNR, Pisa

Responsabile: Fosca Giannotti (CNUCE)

ISI, CNR, Cosenza

Responsabile: Domenico Sacca

Task 3: Librerie scientifiche

Coordinatore: Almerico Murli, CPS-CNR, Napoli

Partecipanti:

CPS, CNR, Napoli

Responsabile: Almerico Murli

CERE, CNR, Palermo

Responsabile: Alessandro Genco

Dipartimento di Informatica, Sistemi e Produzione, Università di Roma Tor Vergata (DIS UniRoma2)

Responsabile: Salvatore Tucci

Dipartimento di Chimica, Università di Perugia, Perugia (Dchim UniPg)

Responsabile: Antonio Laganà

Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria (DEIS UniCal)

Responsabile: Lucio Grandinetti

Task 4: Librerie per l'elaborazione di immagini

Coordinatore: Andrea Clematis, IMA-CNR, Genova

Partecipanti:

IMA, CNR, Genova

Responsabile: Andrea Clematis

Ferrania SpA, Savona

Responsabile: Vittorio Levati

Synapsis srl, Livorno

Responsabile: Riccardo Fontanelli

Task 5: Applicazione in ambito di controllo ambientale e protezione civile

Coordinatore: Giovanni Milillo, ASI, Matera

Partecipanti:

ASI, Matera

Responsabile: Giovanni Milillo, ASI, Matera

IESI, CNR, Bari

Responsabile: Nicola Veneziani

Dipartimento Interateneo di Fisica, Politecnico di Bari (PolibBa)

Responsabile: Luciano Guerriero

Dipartimento di Ingegneria dell'Innovazione, Università di Lecce (ISUFI)

Responsabile: Giovanni Aloisio

Università Di Bari, II Facoltà di Taranto (UniBa2)

Responsabile: Vincenzo Di Lecce

Telespazio SpA, Roma

Responsabile: Ettore Lopinto

Significative collaborazioni esterne:

- SEMA Group SpA, Pont St. Martin (Ao)
- Network Computer House (N.C.H.) SpA, Bologna
- Centro Interuniversitario per il Monitoraggio Ambientale (CIMA), Savona

2. Obiettivo

2.1. Contesto scientifico e tecnologico

Il modello delle architetture a componenti, viste come la logica evoluzione dell'approccio ad oggetti distribuiti, ha come obiettivo la *generalità e la modularità*, la *portabilità ed il riuso* delle applicazioni, *l'interoperabilità* dei linguaggi e *l'integrazione* degli strumenti di sviluppo, nella definizione e realizzazione di applicazioni complesse, spesso di tipo *multidisciplinare*, in modo *indipendente dalle caratteristiche della macchina sottostante* (architettura hardware, sistema operativo, supporti al networking). L'utente deve essere svincolato dalla conoscenza della localizzazione e della realizzazione logica e fisica dei componenti software usati.

Molto critico, e non ancora adeguatamente risolto, è il problema di ottenere, dal modello a componenti, anche *elevate prestazioni* su un ampio range di piattaforme fisiche [1 – 5]. Più precisamente, l'esigenza è quella di *caratterizzare lo sviluppo di applicazioni complesse come applicazioni ad alte prestazioni senza venir meno alle caratteristiche, sopra citate, di indipendenza dalla macchina, in modo che il miglioramento delle prestazioni vada di pari passo con l'evoluzione delle tecnologie di base della macchina e del sottosistema di networking*.

Il problema del raggiungimento di alte prestazioni in sistemi distribuiti è stato finora affrontato in maniera parziale con riferimento solo ad alcuni specifici "colli di bottiglia" del sistema hardware-software, come ad esempio nella realizzazione di server paralleli per la comunicazione o per l'esecuzione di funzioni di compressione e di gestione di masse di dati [11]. Per quanto degne di nota, si tratta di soluzioni "ad hoc" che hanno il serio inconveniente di essere strettamente dipendenti dalla tecnologia della macchina sottostante, e quindi o limitate alla velocizzazione del solo supporto a tempo di esecuzione o, quando attuate al livello delle applicazioni, non integrate nell'ambiente di sviluppo e quindi poco portabili, modificabili e riusabili.

Ribaltando tale visione, l'obiettivo di questa proposta è quello di *integrare* in una stessa piattaforma ITC abilitante complessa le caratteristiche e gli strumenti *dell'elaborazione distribuita, del modello a componenti, e del calcolo ad alte prestazioni* secondo un *approccio unificante*. Dal punto di vista dell'architettura fisica, la piattaforma ITC è costituita da un sistema distribuito, realizzato con tecnologia di networking disponibile, i cui sistemi ospiti includono sistemi paralleli ad alte prestazioni, in generale *eterogenei* l'uno rispetto all'altro. Su tale base si intende realizzare e sperimentare una piattaforma innovativa a tutti i livelli sovrastanti, che recepisca sia le caratteristiche di modularità, riusabilità e portabilità del modello a componenti che quelle di performance del modello ad alte prestazioni.

Rispetto ai progetti citati, questa proposta *non è legata alle sole applicazioni scientifiche* ed affronta il problema dell'integrazione del modello a componenti e del modello ad alte prestazioni da un *punto di vista complessivo ed unificante, esteso a tutti i livelli del sistema*: supporti architetturali e supporti a tempo di esecuzione, strumenti e ambiente di programmazione, componenti di ausilio allo sviluppo di applicazioni (librerie, template, toolkit).

2.2. Approccio e metodologia

L'approccio proposto, oltre che dal punto di vista tecnico, va valutato anche in termini di capacità di coordinare le competenze scientifiche e tecnologiche attorno ad un progetto che produca qualcosa di concreto ed al tempo stesso innovativo, con ricadute sensibili e durature sia sul mondo della ricerca che su quello dell'industria e dell'utenza.

Affinché ciò sia realistico ed adeguato alle risorse in campo, occorre porre alcuni vincoli, in particolare:

- *l'architettura "fisica" è basata fortemente su componenti standard "commodity"* (processori, sistemi operativi, protocolli, strutture di interconnessione, strutture e servizi di networking). In altri termini, la ricerca non prevede lo sviluppo di nuovi componenti hardware-software di basso livello che non possano essere ottenuti, a prestazioni paragonabili, mediante la composizione di componenti commodity. Saranno invece definiti e sviluppati componenti capaci di agire efficacemente da collante, o tali da introdurre elevato valore aggiunto, purché compatibili con le capacità, attuali o previste, della tecnologia italiana ed europea;
- *tutti gli standard software*, attualmente usati per lo sviluppo di applicazioni, devono essere preservati e integrati in un unico ambiente multidisciplinare.

Oltre a workstation e server con architettura uniprocessor o multiprocessor SMP, si considerano macchine parallele di tipo UMA, NUMA, multicomputer e cluster/Beowulf di componenti appartenenti alle classi precedenti. Per tutte queste tipologie di sistema vengono riusati componenti standard a livello di strutture di interconnessione, sistema operativo, sistemi di gestione di basi di dati, linguaggi e strumenti ad oggetti e non, protocolli e servizi di networking.

Fondamentali building block del progetto sono inoltre: *broker e architetture logiche e fisiche a oggetti distribuiti* [8] (CORBA, DCOM, JavaBeans, ActiveX, ecc), *middleware e supporti all'utilizzo di griglie computazionali* [9, 10] (Legion, Globus, ecc).

Il progetto sviluppa *una applicazione significativa* che non solo serva da dimostratore, ma che, nel rispetto dei domini applicativi del Fondo Speciale, abbia un forte impatto sull'utenza nazionale ed europea. Da questo punto di vista, si ritiene che sia prioritario il campo del *controllo ambientale e protezione civile*, che possiede tutti i requisiti, di generalità e necessità di alte prestazioni [14], che devono caratterizzare le nuove piattaforme ITC. L'applicazione è coordinata da ASI.

La piattaforma proposta è strutturata attorno ai seguenti livelli, elencati a partire dalla macchina fisica e salendo gradualmente verso il livello delle applicazioni finali:

- **L0:** *piattaforme fisiche di base*
- **L1:** *supporti ad alte prestazioni*
- **L2:** *ambienti e strumenti di sviluppo applicazioni ad alte prestazioni*
- **L3:** *componenti specializzati di supporto allo sviluppo di applicazioni*
- **L4:** *applicazioni*

Trattandosi di un progetto "verticale", della durata di 2 anni, occorre fissare una metodologia di ricerca che permetta di sviluppare contemporaneamente i diversi livelli. Per il primo anno, le applicazioni verranno realizzate con strumenti esistenti emulando la metodologia che caratterizza i livelli inferiori. Nel secondo anno per le applicazioni verranno utilizzati e sperimentati gli strumenti messi a disposizione, in forma prototipale, dai livelli inferiori.

Di fondamentale importanza, a tutti i livelli, è la capacità di riutilizzare i recenti risultati della ricerca e della tecnologia in questo settore, riconducendoli ad un *contesto unificante e innovativo*.

2.3. Base di partenza

Si ritiene che l'attività sia realisticamente in grado di fornire concrete ricadute scientifico-tecnologiche ed applicative, in quanto

- a) come detto, il sistema è costruito interamente sopra componenti standard "commodity" (processori, macchine parallele e cluster/Beowulf, sistemi operativi, strutture di interconnessione, strutture e servizi di networking e middleware, strumenti e ambienti di sviluppo tradizionali e ad oggetti);
- b) utilizza e assembla, in un contesto unificante ed altamente innovativo, molti risultati e strumenti nati recentemente (progetti europei, CNR, MURST, ASI, ASI-PQE2000);
- c) vede la partecipazione di gruppi da tempo coordinati a livello nazionale sulle tematiche HPC, perseguendo una linea scientifica e tecnologica nel settore dei sistemi general-purpose ad alte prestazioni e loro applicazioni, che, di progetto in progetto, presenta forti e rilevanti evoluzioni, caratterizzazioni e sviluppi sia dal punto di vista scientifico che da quello tecnologico. Il coordinamento è stato reso possibile con la promozione, talvolta con il supporto, degli enti che propongono questa Attività: CNR e ASI.

Segue un elenco di iniziative, nel settore del calcolo ad alte prestazioni, relative a progetti recentemente conclusi, o a progetti in atto, o a proposte ancora oggetto di valutazione, che testimoniano delle competenze esistenti, della vivacità e coordinamento del settore in Italia. Esse sono da considerare sinergiche e funzionali alla presente proposta, in quanto propedeutiche o complementari, in nessun caso sovrapposte ad essa:

- Progetto PQE2000, 1996-99: architetture hardware.-software, ambiente di programmazione parallela SkIE, collaborazione con il progetto Eureka HPPC/SEA, collaborazione con il Ministro delle Finanze sul tema della lotta all'evasione fiscale con strumenti di calcolo ad alte prestazioni,
- Progetto cofinanziato Murst "MOSAICO", 1996-98,
- Progetto cofinanziato Murst "Algoritmi per grandi insiemi di dati: Scienza e Ingegneria", 1999,
- Progetto ASI-PQE2000, 2001 – 2003, "Sviluppo di applicazioni di Osservazione della Terra mediante sistemi e strumenti di Calcolo ad Alte Prestazioni" (partecipazione industriale di Telespazio SpA e Synapsis),
- Proposta CNR, Agenzia 2000, "Ambiente per lo sviluppo di applicazioni ad alte prestazioni multiplatforma e multilinguaggio, basato sul modello ad oggetti e sulla programmazione parallela strutturata",
- Proposta CNR, Agenzia 2000, "Griglie Computazionali e Applicazioni",
- Proposta CNR, Fondo Speciale 1999, "Tecnologie per arricchire e fornire accesso a contenuti - linea Mine the WEB",
- Proposta CNR, Fondo Speciale 1999, "Grid Computing: Tecnologie Abilitanti e Applicazioni per eScience".

Nell'ambito delle iniziative citate sono state sviluppate, oltre che competenze di alto livello nel settore, anche tecnologie e risorse di assoluta utilità per l'avvio della ricerca e della sperimentazione. Ne sono esempi:

- configurazioni di tipo cluster o Beowulf presso quasi tutti i vari istituti e dipartimenti,
- macchine massicciamente parallele presso CPS-CNR, ISUFI, CNUCE, IESI, IFCAI, Univ. Perugia,
- strumenti e ambienti di programmazione parallela e Grid presso Università di Pisa, CPS-CNR, ISI, ISUFI, CNUCE, IFCAI, CERRE, Univ. Perugia, Univ. Venezia,
- ambienti, strumenti e librerie di calcolo scientifico presso CPS-CNR, CNUCE, ISUFI, Univ. e CSCISM di Perugia, Univ. Roma Tor Vergata, Politecnico di Milano,
- ambienti, strumenti e librerie di KDD e Data Mining presso ISI, Univ. Pisa, CNUCE,
- strumenti e librerie di elaborazione di immagini, GIS, osservazione della terra presso ASI Matera, IMA, IESI, Politecnico di Bari, IFCAI, CNUCE.

3. Tematiche e strutturazione del progetto in task

L'attività si articola nei seguenti task.

Task 1: Ambiente di programmazione portatile a componenti parallele

Il prodotto consiste nell'ambiente per la programmazione di applicazioni HPC in modo indipendente dall'architettura fisica, integrando in un contesto parallelo ambienti e strumenti standard sequenziali e distribuiti. La metodologia è basata sull'integrazione di programmazione parallela strutturata e programmazione a componenti. È nel supporto, compilativo e run-time, che vengono sviluppati componenti per l'ottimizzazione delle prestazioni in modo adattabile alla macchina sottostante. Questa è costituita da sistemi a oggetti distribuiti come CORBA, DCOM, JavaBeans, eventuale supporti a Grid (Globus, Legion), con nodi paralleli.

Task 2: Componenti ad alte prestazioni per Data/Web Mining e motori di ricerca

Questo task realizza, sopra il livello del Task 1, ambienti specializzati costituiti da componenti paralleli finalizzati alla estrazione e ricerca di informazione e conoscenza da fonti strutturate e semi-strutturate (Web), nonché all'utilizzo dell'informazione estratta per question answering e problem solving in generale. Sono previsti i seguenti prodotti integrati:

- ambiente per Knowledge Discovery Process
- ambiente per Question Answering
- motori di ricerca per dati semistrutturati.

Task 3: Librerie scientifiche

Il prodotto è un insieme ampliabile di componenti (toolkit) di supporto per applicazioni scientifiche, che consenta l'utilizzazione di algoritmi numerici e software di base sofisticati e aggiornabili. Il toolkit, che verrà integrato nell'ambiente del Task 1, contiene in particolare:

- sistemi lineari sparsi e di grandi dimensioni;
- calcolo di autovalori e autovettori di matrici sparse e di grandi dimensioni;
- trasformate veloci e preconditionatori wavelet.

Task 4: Librerie per l'elaborazione di immagini

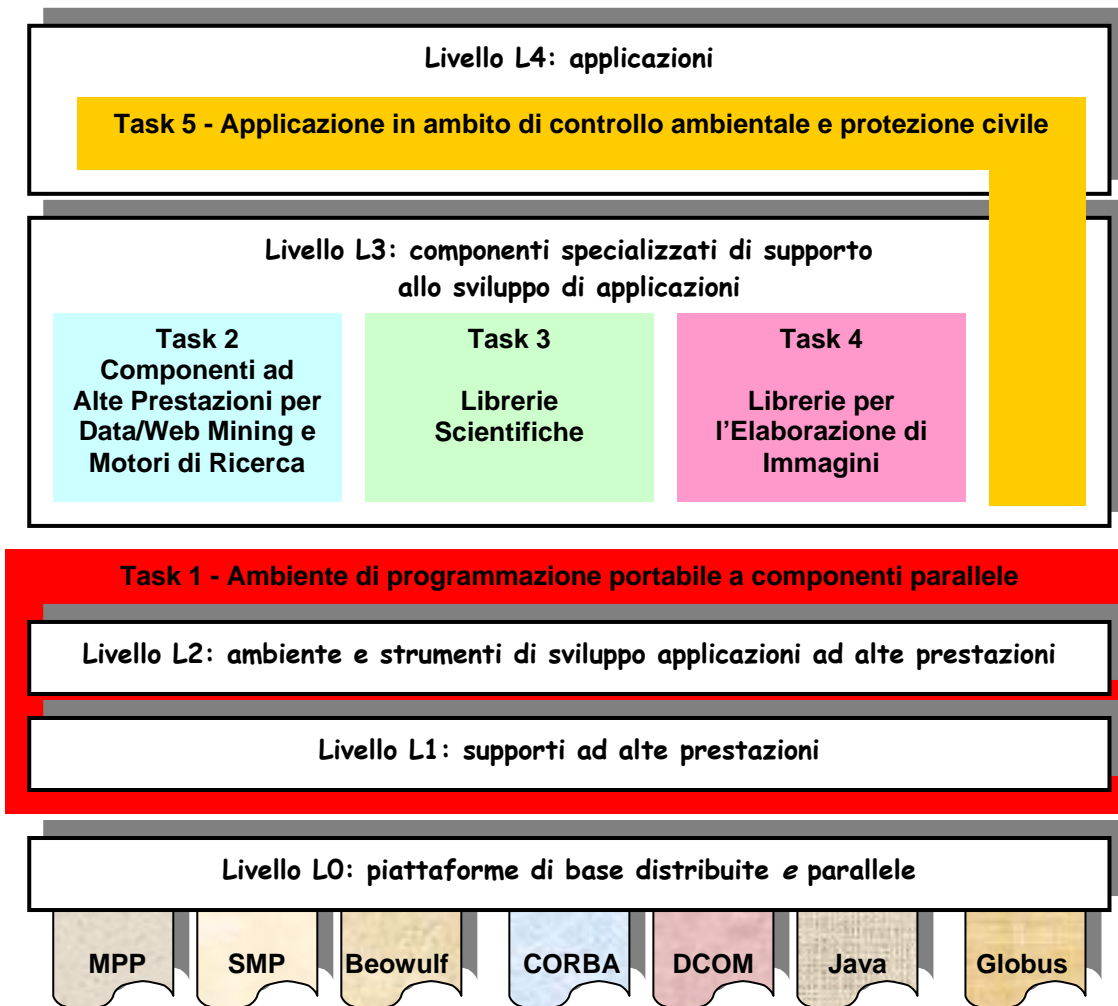
Il prodotto è una libreria parallela, sviluppata con la metodologia del Task 1, per l'elaborazione di immagini ad altissime prestazioni, secondo le seguenti funzionalità: acquisizione di immagini di natura eterogenea (da fonti differenti ed in formati differenti), trattamento e analisi, compressione, conversione del formato delle immagini, ricerca delle immagini (totali o parziali). La libreria verrà sperimentata e dimostrata in applicazioni biomedicali.

Task 5: Applicazione in ambito di controllo ambientale e protezione civile

L'applicazione consiste nel processamento SAR interferometrico nei suoi due principali aspetti: la generazione del DEM (Digital Elevation Model) per la mappatura delle altezze, e la rilevazione di movimenti del terreno con accuratezze dell'ordine del centimetro, anche su lunghi intervalli di tempo. I dati sono quelli acquisiti dalla Shuttle nell'ambito della Terza Missione SRTM, esteso ai dati ERS. Si tratta di una catena di processamento già operativa in ASI in ambiente sequenziale, della quale verranno parallelizzati i moduli più critici agli effetti delle prestazioni per permettere l'effettivo tempo reale nello sfruttamento dei dati nei servizi civili, come lo studio delle frane.

Task 6: Documentazione e divulgazione dei risultati

La figura seguente illustra la relazione tra i task ed i livelli della piattaforma:



4. Task 1: Ambiente di programmazione portabile a componenti parallele

4.1. Obiettivo generale

Obiettivo generale di questo Task e' quello di mettere a disposizione due livelli (L1: supporto, L2: ambienti e strumenti di sviluppo) che forniscano allo sviluppatore di applicazioni un ambiente omogeneo per la realizzazione di applicazioni caratterizzate da alte prestazioni ma sviluppate con metodologie sostanzialmente indipendenti dalle caratteristiche tecnologiche dell'architettura sottostante, sia in termini di componenti (tipi di macchine, prestazioni dei processori, etc.) che della rete di interconnessione (banda, latenza, etc.).

In particolare, si vuole realizzare un framework che permetta di sviluppare applicazioni secondo una metodologia integrata sostanzialmente derivata da modelli diversi e originariamente distinti:

- modelli sviluppati esplicitamente per l'elaborazione distribuita
- modelli per la programmazione parallela strutturata
- modelli a componenti, con particolare enfasi sulle tematiche del riuso del software e della modularità

La metodologia di sviluppo delle applicazioni ad alte prestazioni supportata dai livelli L1 ed L2 dovrà quindi avere proprietà quali:

- permettere il riuso di software sequenziale preesistente, con minime modifiche (incapsulamento del sw esistente in strutture/moduli wrapper)
- permettere lo sviluppo di applicazioni caratterizzate da alte prestazioni *indipendentemente* dalle caratteristiche della macchina (o delle macchine) su cui verranno fatte girare
- permettere lo sviluppo di applicazioni con caratteristiche di alte prestazioni che possano essere utilizzate come componenti (preconfezionati) di altre applicazioni, sia ad alte prestazioni che sequenziali
- mettere a disposizione del programmatore una serie di strumenti standard che permettano di sviluppare parti di applicazioni che utilizzano forme standard di parallelismo in maniera semplice e sintetica.

Questi obiettivi verranno perseguiti progettando e realizzando i due livelli in maniera congiunta, tenendo conto di tutte le esperienze sviluppate dai membri delle unità operative che partecipano al progetto nell'ambito di progetti attualmente in corso o precedentemente condotti che risultano sinergici con gli obiettivi di questo progetto.

4.2. Livello L1: supporti ad alte prestazioni

Gruppi: DI UniPi (Danelutto, Baiardi, Vanneschi), Synapsis (Fontanelli, Guerri), IFCAI (Machi), DI UniVe (Orlando, Balsamo), ISTI (Laforenza), DEI PoliMi (Serazzi, Cremonesi)

Il livello L1 mette a disposizione tutta una serie di meccanismi che possono essere utilizzati per la realizzazione di un ambiente di programmazione ad alte prestazioni strutturato secondo il paradigma di programmazione ad oggetti. In particolare, il livello L1 metterà a disposizione meccanismi per:

- la realizzazione di reti di processi/thread
- lo scheduling di processi/thread sui nodi dell'architettura fisica
- la comunicazione fra processi/thread (punto a punto o collettiva)
- la condivisione di informazione/dati fra processi/thread
- la realizzazione di template (ovvero di forme di cooperazione predefinite, parallele e/o distribuite ed efficienti)
- la fault tolerance e la sicurezza

il tutto con caratteristiche che permettano lo sfruttamento di questi meccanismi per lo sviluppo di applicazioni ad alte prestazioni su architetture che vanno dal cluster di macchine omogenee alla rete di macchine disomogenee distribuite su scala geografica.

Per realizzare il livello L1 si ricorrerà all'utilizzo di supporti software/middleware esistenti e alla loro integrazione in un ambiente uniforme, in modo da riutilizzare tutta l'esperienza precedentemente acquisita in questo campo sia dalle unità che fanno parte del progetto che da altri gruppi di ricerca.

I meccanismi messi a disposizione dal livello L1 potranno essere utilizzati in due forme, all'interno del progetto:

- come tali, per lo sviluppo di prototipi di applicazioni e/o librerie ad alte prestazioni secondo diversi paradigmi di programmazione
- come meccanismi di una macchina astratta su cui compilare programmi paralleli e/o distribuiti mediante gli strumenti messi a disposizione dal livello L2.

La prima possibilità verrà in particolare sfruttata dalle unità operative che all'interno di questo progetto svilupperanno prototipi di applicazioni nella prima fase del progetto, cioè in quella fase in cui ancora non saranno pronti gli strumenti del livello L2.

4.3. Livello L2: ambiente e strumenti

Gruppi: DI UniPi (Danelutto, Vanneschi), Synapsis (Fontanelli, Guerri), DI UniVe (Orlando)

Il livello L2 mette a disposizione ambienti e strumenti per la programmazione di applicazioni ad alte prestazioni secondo una metodologia basata su tecniche di programmazione parallela strutturata e a componenti. Gli strumenti garantiscono la possibilità di sviluppare applicazioni parallele ad alte prestazioni in maniera sostanzialmente indipendente dalle caratteristiche della macchina utilizzata per l'esecuzione di tali applicazioni e/o dalla sua struttura fisica (macchina con nodi di elaborazione eterogenei o omogenei, supporto per condivisione di dati (e/o I/O) primitivo o fornito mediante middleware/software, etc.).

In particolare gli strumenti mettono a disposizione:

- un modello di programmazione strutturato
 - basato sulla tecnologia ad oggetti
 - e su un modello di linguaggio di coordinamento
 - che mette a disposizione come primitive tutte le forme di parallelismo comunemente utilizzate per la realizzazione di applicazioni parallele ad alte prestazioni
 - che permette il riutilizzo di componenti software pre-esistenti, sia di tipo sequenziale che parallele (se sviluppate con gli stessi strumenti)
 - che permette di rendere disponibili le applicazioni sviluppate come componenti da riutilizzare in altri sistemi software, per esempio come server che mettono a disposizione servizi ad alte prestazioni
- un insieme di strumenti di compilazione che
 - permettono di compilare programmi/componenti parallele strutturate sui meccanismi definiti nel livello L1
 - ottimizzano il codice oggetto in funzione delle caratteristiche della macchina su cui tale codice dovrà essere fatto girare in maniera sostanzialmente trasparente al programmatore
 - implementano tutte le politiche di sicurezza necessarie per preservare l'integrità e la sicurezza dei dati coinvolti nelle computazioni
 - rendono disponibili al programmatore, in modo strutturato ed uniforme tutta una serie di meccanismi tipicamente utilizzati per la realizzazione di applicazioni parallele quali ad esempio:
 - a. accesso a strutture dati (oggetti) condivisi
 - b. accesso ad oggetti definiti esternamente all'ambiente di programmazione (e.g. via CORBA, RMI, RPC, etc.)
 - c. accesso a file system paralleli/distribuiti per il supporto di computazioni parallele I/O intensive
- un run time che permette di far girare applicazioni realizzate secondo il modello di programmazione proprio di questo livello su una serie di architetture hardware/software diverse; per esempio:
 - su cluster di macchine omogenee, sia in termini di processore che di sistema operativo
 - su macchine parallele SMP
 - su macchine parallele a memoria distribuita
 - su reti di macchine anche disomogenee fra di loro (in termini di processore, memoria, sistema operativo) e distribuite su scala non necessariamente locale

Questo ambiente di programmazione sarà reso disponibile, entro la prima fase del progetto, a tutte quelle unità che partecipano alla realizzazione di (prototipi di) applicazioni dimostrative nel progetto.

4.4. Base di partenza

La base di partenza è rappresentata dai prodotti della ricerca di PQE2000 sui temi della tecnologia software [12, 13]:

- ambiente di sviluppo SkIE,
- macchina virtuale intermedia a scambio di messaggio + memoria condivisa,

e dai prodotti emergenti dal progetto ASI-PQE2000, in particolare l'ambiente ASSIST.

5. Task 2: Componenti ad alte prestazioni per Data/Web Mining e motori di ricerca

Le attività descritte in questa sezione hanno come obiettivo comune la costruzione di ambienti o componenti di ambienti finalizzati alla estrazione e ricerca di informazione e conoscenza da fonti strutturate e semi-strutturate (Web), nonché all'utilizzo dell'informazione estratta per Question Answering e Problem Solving in generale.

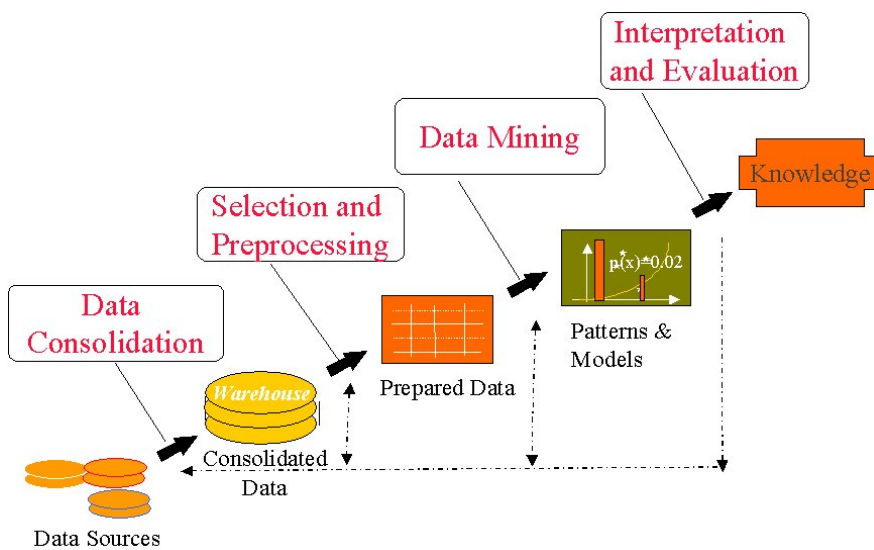
Nel contesto generale del progetto tutte le attività mirano a sfruttare lo strato sottostante o per realizzare in modo parallelo specifici algoritmi (ad esempio, specifici algoritmi di data mining o specifiche componenti dei motori di ricerca) o per produrre componenti progettate in modo da essere integrabili in un sistema parallelo a componenti.

Le sottosezioni successive descrivono per ciascuna componente/ambiente proposto il suo inquadramento nel contesto internazionale e la base tecnologia di partenza disponibile ai gruppi, e i suoi obiettivi.

5.1. Ambiente per Knowledge Discovery Process

Gruppi: DI Unipi (Turini, Pedreschi), CNUCE (Giannotti)

Il processo di estrazione della conoscenza (KDD process) è costituito da una serie di fasi che precedono e seguono il data mining, illustrate nella figura sottostante. Lo sviluppo di applicazioni complesse, dal *market basket analysis* alla *fraud detection* fino al *Web mining*, richiede la possibilità di verticalizzare tutte tali fasi adattandole ed integrandole dentro un decision support system, oppure un sistema di Web management, un motore di ricerca o un portale per commercio elettronico.



Il Pisa KDD Laboratory, a cui collaborano sia ricercatori del Dipartimento di Informatica che del CNUCE, ha sviluppato un ambiente a supporto del processo di data mining in cui i risultati intermedi, gli input e gli output agli algoritmi di data mining, le azioni di manipolazione dei dati e lo stesso query language sono rappresentati in notazione XML. Il sistema così concepito presenta da un lato l'interoperabilità tra vari tool di data mining, nonché la possibilità di esprimere la loro composizione e dall'altro è predisposto all'applicazione degli stessi strumenti su dati codificati in XML. Tale ambiente può costituire una base di partenza per questo obiettivo del progetto. Gli obiettivi sono da un lato l'integrazione di algoritmi di data mining paralleli sviluppati da ISI Cosenza e Dipartimento di Informatica e dall'altro la riprogettazione del sistema in modo tale che sia inseribile in modo smooth in un contesto di sistema parallelo a componenti.

Nell'ambito della progettazione e sviluppo di algoritmi di data mining paralleli, l'obiettivo è quello di sviluppare algoritmi che siano scalabili e in grado di analizzare grandi moli di dati. In particolare si intendono utilizzare paradigmi innovativi come quelli forniti dalla programmazione evolutiva e dalle forme di vita artificiale (boids, flocks, ants etc.) per definire algoritmi di data mining per la classificazione e il clustering spaziale di dati.

Più specificamente, per ciò che riguarda gli algoritmi di classificazione l'approccio che si intende seguire è quello di utilizzare la programmazione evolutiva per far evolvere alberi di decisione. In tale ambito recenti lavori svolti dall'ISI hanno mostrato che l'utilizzo della programmazione evolutiva rappresenta un metodo alternativo per la costruzione di alberi di decisione più semplici e accurati rispetto a quelli generati ad esempio da strumenti commerciali come C4.5. L'ISI ha già sviluppato uno strumento parallelo per la programmazione evolutiva basato sul modello a diffusione. Tale strumento può costituire il punto di partenza per la realizzazione degli obiettivi del progetto.

Per quanto riguarda il problema del clustering spaziale, l'idea è quella di utilizzare comportamenti e strategie di alcune forme di vita artificiale per progettare algoritmi di clustering da utilizzare per l'analisi spaziale dei dati. In particolare, l'approccio riguarda l'utilizzo di agenti di tipo ant o flock da utilizzare per costruire una strategia smart di ricerca

capace di eseguire una analisi esplorativa spaziale dei dati per scoprire i cluster presenti. Attualmente disponiamo di un primo prototipo sequenziale dell'algoritmo che è in fase di valutazione e che può costituire la base per lo sviluppo di un algoritmo parallelo da integrare nel sistema di data mining parallelo che si intende costruire nel progetto.

5.2. Ambiente per Question Answering

Gruppo: DI UniPi (Attardi)

L'obiettivo delle tecniche di Question Answering è di andare oltre le capacità degli attuali sistemi di Information Retrieval, che sono efficaci nell'individuare i documenti rilevanti un certo argomento, per riuscire a fornire direttamente all'utente le risposte ad un quesito.

Per raggiungere questo obiettivo, si sono combinate tecniche di Information Retrieval e di Information Extraction. Le conoscenze contenute nei documenti da interrogare sono estratte sotto forma di concetti e di relazioni tra loro. Queste conoscenze vengono archiviate sotto forma di frasi con opportuni tag, in una base di conoscenze costituita da un sistema di IR, indicizzate sulla base dei concetti. I concetti sono espressi sotto forma di coppie [entità, categoria semantica]. Le entità vengono estratte dai testi tramite una serie di elaborazioni che comprendono un Sentence Splitter, un POS Tagger, un Thesaurus (WordNet). Le entità vengono poi classificate all'interno di un'opportuna ontologia da un Semantic Tagger.

In fase di interrogazione, si estraggono dalla domanda i concetti e si estraggono dalla base di conoscenze i documenti più pertinenti. Questi vengono suddivisi in paragrafi e di ognuno viene stimata la pertinenza con la domanda. La pertinenza tiene conto di aspetti semantici della domanda, quali il tipo della domanda, il focus ed il tipo della risposta richiesta.

Con il prototipo PISAB, l'unità del Dipartimento di Informatica di Pisa ha partecipato alle valutazioni dei sistemi di Question Answering della conferenza TREC-9, ottenendo risultati positivi.

Come sviluppi del sistema di Question Answering si prevede di raffinare la capacità di individuare relazioni tra concetti presenti nei documenti Web, es. relazioni di sussunzione, causa e temporali. Aggiungendo al sistema capacità deduttive su queste relazioni, sarà possibile rispondere anche a interrogazioni non triviali, che richiedono di utilizzo di relazioni e conoscenze estratte da più documenti.

Le tecniche sviluppate nel progetto possono essere utilizzate nel settore dei *personal assistants*. La capacità di apprendere relazioni e concetti è necessaria per migliorare strumenti di assistenza personale, in modo che si possano adattare al vocabolario ed alle esigenze dell'utente. Un'applicazione possibile è un assistente personale per la categorizzazione di bookmarks, della posta o di altre collezioni personali di documenti Web.

Anche in questo caso gli sviluppi previsti saranno concepiti in modo tale da rendere il sistema inseribile naturalmente in un ambiente parallelo a componenti.

5.3. Motori di Ricerca per dati semi-strutturati

Gruppo: DI UniPi (Luccio, Ferragina)

Lo sviluppo di strutture dati e algoritmi efficienti per problemi di ricerca su grandi collezioni testuali riveste oggi un ruolo strategico per due motivi essenziali: da una parte si osserva un aumento del gap tra le prestazioni dei processori e le velocità di accesso alle memorie, dall'altra si nota una crescita esponenziale nel numero e dimensione dei documenti reperibili in forma elettronica che rende sempre più inevitabile l'accesso ai dispositivi di memoria di massa. Numerosi sforzi scientifici sono oggi pertanto proiettati verso la definizione di motori di ricerca che offrano non soltanto migliori tempi di risposta alle query poste dall'utente, ma anche una riduzione dello spazio occupato dalle strutture dati utilizzate.

Questo scenario è reso ancora più interessante dall'affermarsi di *linguaggi di marcatura*, quali SGML e XML, che consentono di *strutturare* i documenti arricchendoli con informazioni aggiuntive di vario tipo: categorizzazione di porzioni distinte (ad esempio per identificare sinonimi, linguaggi, citazioni, integrazioni, ecc.), suddivisione tipografica (cartelle, volumi, pagine, paragrafi, ecc.) o tipologica (versi, prosa, dialoghi, ecc.). La disponibilità sempre più crescente di documenti strutturati, nonché la ricchezza di informazioni ausiliarie in essi contenute, rende quanto mai interessante lo studio e il progetto di motori di ricerca che possano auspicabilmente avvantaggiarsi di esse per supportare "ricerche semantiche" più selettive, efficaci ed efficienti di quelle attuali. Va da sé che i sistemi di *Information Retrieval* classici per quanto sofisticati ed efficienti, presentano delle apparenti limitazioni se utilizzati su documenti (semi)strutturati in quanto questi non possono essere semplicemente visti come sequenze "piatte" di parole (*word-based index*). Per ovviare a questo inconveniente alcune grandi case produttrici di software si sono indirizzate verso la definizione di strumenti di ricerca per documenti XML basati su DBMS, ma queste soluzioni hanno immediatamente manifestato delle profonde

limitazioni sia dal punto di vista delle prestazioni in tempo che in spazio. E' oggi sempre più evidente che uno strumento di indicizzazione per questo tipo di documenti deve essere progettato tenendo direttamente conto delle loro peculiarità strutturali le quali devono essere incapsulate a priori all'interno della struttura dati di indicizzazione, e non sfruttate a posteriori nel processo di interrogazione (come avviene peraltro oggi in gran parte dei motori di ricerca per XML/SGML).

L'unità del Dipartimento di Informatica di Pisa ha già sviluppato un prototipo di motore di ricerca per documenti puramente testuali che racchiude in sé una caratteristica innovativa: è un indice compresso che non richiede la decompressione completa dell'archivio all'atto della ricerca. I risultati preliminari di questo studio sono stati pubblicati sui *proceedings* della conferenza internazionale *IEEE Focs 2000* [25, 26]. La struttura dati per l'indicizzazione da noi proposta, e denominata *FM-index*, trae vantaggio dalla comprimibilità dei testi per ridurre lo spazio occupato, senza però pregiudicare in alcun modo l'efficienza delle ricerche eseguibili su di essa. Più precisamente lo spazio totale è funzione lineare dell'entropia dell'insieme dei dati indicizzato, e quindi risulta ottimo nel senso della teoria dell'informazione; inoltre, la complessità in tempo delle ricerche è del tutto paragonabile a quella ottenuta dai migliori indici testuali sia per quanto concerne il conteggio delle occorrenze che il recupero delle stesse, se siamo in presenza di query selettive.

In questo progetto ci proponiamo di investigare l'applicabilità dell' *FM-index* al progetto di sofisticati motori di ricerca per documenti (*semi*)strutturati. L'idea è quella di sfruttare la compattezza ed efficienza dell' *FM-index* per arricchire il testo strutturato SGML/XML di *ulteriori informazioni* le quali consentano di supportare ricerche efficienti pur mantenendo comunque contenuto lo spazio totale occupato. Queste informazioni aggiuntive verranno inoltre utilizzate per permettere ricerche "complesse" che tengano conto della struttura del testo, e per consentire analisi sofisticate della struttura dello stesso da parte degli utenti.

Il progetto di questo motore di ricerca e il suo sviluppo saranno concepiti in modo tale da rendere il sistema inseribile in un ambiente parallelo a componenti. A ciò si accompagnerà uno studio sulle problematiche legate alla ricerca di informazioni in ambienti paralleli e distribuiti [24], campo di indagine questo quanto mai interessante ma povero di risultati significativi fino ad oggi.

6. Task 3: Librerie scientifiche

Gruppi: CPS (Murli, D'Ambra, Di Serafino), CERE (Genco, Tortorici), DIS UniRoma2 (Filippone, Tucci), Dchem UniPg (Laganà, Tarantelli)

6.1. Contesto scientifico e tecnologico

E' noto come la strategia di sviluppo del software per applicazioni scientifiche, che tende a realizzare obiettivi come il riuso, la portabilità, l'estendibilità, la modularità, la robustezza, l'efficienza, sia basata sull'uso delle librerie in termini di "building block". Attualmente, alla luce dell'evoluzione nell'ambito delle architetture hardware e software, basata sull'approccio "a componenti", è necessario ripensare alle strategie di sviluppo di tali librerie allo scopo di preservare le caratteristiche sopra esposte in contesti ampi di interoperabilità di macchine, linguaggi, strumenti di sviluppo, etc.

Su tale tematica è stata incentrata l'ultima "Working Conference" organizzata dal "Working Group" dell'IFIP sul Software Numerico [<http://www.ifip.or.at/>], e la realizzazione di "toolkit" numerici per il Calcolo Scientifico ad Alte Prestazioni, basati sull'approccio a componenti, è l'obiettivo di numerosi progetti connessi al "Common Component Architecture Forum" americano [<http://www.acl.lanl.gov/cca-forum/>]. Tra questi, i progetti ALICE di Argonne [<http://www-unix.mcs.anl.gov/alice/>], LSA dell'Indiana University [<http://www.extreme.indiana.edu/pseware/lsa/>], NetSolve dell'Università del Tennessee [<http://www.cs.utk.edu/netsolve/>].

La strategia che si va delineando sembra essere quella di definire e sviluppare componenti software composti da una o più implementazioni (ottimizzazioni "machine-dependent") di algoritmi numerici, e da opportune interfacce (API, Application Programming Interface), che consentano astrazioni matematiche per individuare classi di solutori (ad es. sistemi lineari sparsi, sistemi lineari densi, ottimizzazione, trasformate, ...), piuttosto che particolari algoritmi e strutture dati [27].

6.2. Obiettivi e metodologia

Obiettivo fondamentale del task è la realizzazione di un insieme ampliabile di componenti (toolkit) di supporto per applicazioni scientifiche, che consenta l'utilizzazione di algoritmi numerici e software di base sofisticati e "up-to-date" in maniera il più possibile trasparente ed efficiente su un'ampia gamma di architetture ad alte prestazioni attualmente disponibili.

Tali componenti saranno sviluppati secondo una metodologia per cui moduli software paralleli per ambienti a memoria distribuita e/o a memoria condivisa, sviluppati con strumenti standard e di provata affidabilità, quali le librerie di scambio di messaggi come MPI ed i linguaggi come il Fortran ed il C, con le eventuali estensioni (ad es. OpenMP), vengano integrati in un ambiente complessivo che ne garantisca il massimo grado di riusabilità e di interoperabilità, anche su piattaforme distribuite. Questo comporta, da un lato, la necessità di tenere presenti le esigenze computazionali di applicazioni concrete su larga scala, in modo da realizzare strumenti adeguati e di sicura utilità, dall'altro, la necessità di interagire con gli sviluppatori dell'ambiente complessivo per una definizione chiara e robusta delle interfacce tra i componenti.

Il toolkit per le applicazioni scientifiche conterrà componenti per la risoluzione dei seguenti problemi, che costituiscono il nucleo computazionale di numerose applicazioni tecnico-scientifiche:

- sistemi lineari sparsi e di grandi dimensioni;
- calcolo di autovalori e autovettori di matrici sparse e di grandi dimensioni;
- trasformate veloci e preconditionatori wavelet.

Alcuni dei componenti suddetti saranno sperimentati nell'ambito di uno specifico dominio applicativo (chimica quantistica), utilizzandoli nella simulazione su base molecolare di alcuni processi chimici. Ciò servirà a valutare la metodologia generale utilizzata per l'integrazione dei componenti nell'ambiente complessivo e fornirà indicazioni per l'introduzione di eventuali modifiche ed ottimizzazioni.

Il progetto sarà organizzato secondo le seguenti fasi:

- individuazione di algoritmi e software, in possesso dei partecipanti o di pubblico dominio, per la costruzione dei singoli componenti;
- implementazioni e/o ottimizzazione degli algoritmi e del software individuato, secondo i paradigmi di programmazione parallela e gli strumenti standard;
- definizione delle specifiche relative all'interfaccia con l'ambiente complessivo, in accordo con le funzionalità di quest'ultimo;
- implementazione delle interfacce;
- sperimentazione e valutazione dei risultati.

6.3. Punto di partenza

Il punto di partenza per le attività sopra descritte sarà costituito dai seguenti prodotti software, già sviluppati, in altri contesti, dai partecipanti al progetto:

- PSBLAS, un package per l'algebra lineare su matrici sparse e di grandi dimensioni [28];
- PB-LANCZOS, un package per il calcolo di autovalori/autovettori di matrici sparse e di grandi dimensioni [29]
- codici di chimica quantistica, utilizzati per la simulazione di reazioni di interesse ambientale [30, 31] .

Le attività relative al Task saranno suddivise tra le varie Unità Operative coinvolte come di seguito specificato.

- Coordinamento delle attività: *CPS-CNR*;
- sviluppo di componenti per la risoluzione di sistemi lineari sparsi e di grandi dimensioni: *DIS UniRoma2*;
- sviluppo di componenti per il calcolo di autovalori e autovettori di matrici sparse e di grandi dimensioni: *CPS-CNR*;
- sviluppo di componenti per trasformate veloci e preconditionatori wavelet: *CERE-CNR*;
- sperimentazione di alcune delle componenti in codici applicativi di chimica quantistica: *DChem UniPg*.

7. Task 4: Librerie per l'elaborazione di immagini

7.1. Contesto scientifico e tecnologico

Il trattamento di immagini digitali è uno degli aspetti più rilevanti per lo sviluppo di applicazioni biomediche avanzate ed in altri settori. Un primo aspetto importante è quello dell'acquisizione dell'immagine e della sua interpretazione. Nel settore medico le tecnologie di rilievo sono ad esempio la tomografia computerizzata e la risonanza magnetica. Un ulteriore aspetto è l'archiviazione dell'immagine in un formato opportuno, la conversione tra formati, l'organizzazione di data base di immagini, e la consultazione degli stessi in linea.

Per ognuna di queste attività sono disponibili molti algoritmi, strumenti software e librerie che costituiscono una buona base di partenza per lo sviluppo di applicazioni [20]. Un problema noto è l'alto costo computazionale degli algoritmi utilizzati, che spesso si traduce nell'impossibilità di sviluppare l'applicazione di interesse, in particolare modo quando questa è soggetta a vincoli di tempo reale ed utilizza grandi quantità di dati. Per ovviare, a questo problema si sono sviluppate molte ricerche mirate all'utilizzo dell'elaborazione parallela e di tecniche di calcolo ad alte prestazioni. Queste ricerche hanno condotto allo sviluppo di algoritmi paralleli ed hanno quindi chiarito che è possibile sviluppare strumenti efficaci ed efficienti, ma la disponibilità di questi in ambienti di produzione è per il momento solo parziale. A questo proposito tra i problemi che rimangono da superare possiamo ricordare:

- sviluppo di algoritmi paralleli portabili in modo efficiente su diverse architetture di calcolo;
- possibilità di utilizzare risorse di calcolo eterogenee, scegliendo architetture avanzate o architetture a basso costo in base a criteri che considerano sia i requisiti delle applicazioni sia le risorse economiche disponibili;
- accesso alle funzioni richieste tramite una libreria che mascheri all'utente finale (lo sviluppatore di applicazioni e l'utilizzatore delle stesse) la complessità dell'elaborazione parallela.

Lo sviluppo di tecniche di grid computing e di metodologie di programmazione parallela strutturata che consentono di ottenere programmi portabili ed efficienti, combinati con la grande disponibilità di algoritmi paralleli nei settori di interesse rappresentano la premessa per lo sviluppo di librerie parallele pronte ad essere integrate nello sviluppo di applicazioni ... utilizzabili in diversi contesti quali la ricerca, gli enti di servizio, le industrie.

7.2. Obiettivo e metodologia

L'obiettivo è quello di progettare e produrre una libreria parallela in grado di mettere a disposizione dello sviluppatore di applicazioni biomediche funzioni per l'elaborazione di immagini rispondenti a criteri di alte prestazioni. Questa libreria sarà portabile su architetture eterogenee in modo efficiente e dovrà nascondere allo sviluppatore delle applicazioni ed agli utilizzatori delle stesse, la complessità di utilizzo di risorse distribuite di calcolo ad alte prestazioni. Infine dovrà essere possibile utilizzare la libreria per lo sviluppo di applicazioni complesse ed eventualmente integrate in sistemi proprietari.

Si intendono sfruttare le funzionalità messe a disposizione dai livelli L1 ed L2 per sviluppare la libreria tenendo conto che questa sarà organizzata in diverse componenti di base, o building block, e conterrà moduli specializzati dal punto di vista funzionale.

I servizi messi a disposizione dal livello L1 riguarderanno prima di tutto le funzioni di mapping e load balancing, consentendo quindi di fornire ad ogni applicazione la potenza di calcolo di cui questa ha bisogno.

L'organizzazione in componenti sfrutterà le funzionalità definite in L2 che permettono di sviluppare codice parallelo strutturato e richiamerà in parte quella di ben note librerie computazionali quali Scalapack:

- uno o più building block conterranno i kernel computazionali sequenziali di base;
- un building block conterrà le funzionalità di comunicazione, partizionamento dei dati e strutturazione parallela;
- uno o più building block conterranno i moduli paralleli (realizzati utilizzando gli elementi di 1 e 2) per lo sviluppo delle applicazioni.

I kernel sequenziali e i moduli paralleli che costituiscono l'interfaccia per lo sviluppo delle applicazioni saranno specializzati per funzionalità, quali ad esempio, il trattamento di immagini su base puntuale, areale, o nel dominio delle frequenze, la conversione tra formati ed altro.

Queste funzionalità saranno realizzate a partire da algoritmi già esistenti ed utilizzati in contesti affini; eventualmente attraverso un'opportuna modifica e/o adattamento ai dettami architetture del progetto. Inoltre eventuali algoritmi ad hoc potranno essere definiti in caso di vincoli (es. di natura hardware da parte del sistema/dispositivo sorgente dell'immagine) e/o requisiti specifici dell'applicazione.

I moduli funzionali potranno comunicare tra di loro ed interfacciarsi con altre librerie (ad esempio quelle sviluppate in nel contesto del data mining - web mining) permettendo quindi lo sviluppo di applicazioni verticali e complesse.

Il building block preposto alla comunicazione ed alla gestione del parallelismo sarà costruito utilizzando le primitive e gli oggetti messi a disposizione dal livello L2 e realizzando una loro personalizzazione per gli algoritmi considerati. A questo proposito è importante notare che nel settore considerato, molti degli algoritmi di calcolo hanno caratteristiche comuni così riassumibili:

- dominio dei dati rappresentato in molti casi da singoli pixel e/o griglie con struttura regolare;
- località della computazione (sul singolo pixel o su un'area);
- possibilità di partizionamento statico del carico utilizzando opportuni modelli di costo.

Ovviamente bisognerà considerare accanto a queste possibilità l'utilizzo di strutture dati e modelli più complessi, la possibilità di avere algoritmi non locali, e la necessità di un bilanciamento dinamico del carico.

7.3. Punto di partenza

L'Istituto per la Matematica Applicata (IMA) di Genova del Consiglio Nazionale delle Ricerche ha ampie competenze in settori di interesse per il progetto proposto. In particolare presso l'IMA sono state sviluppate attività di ricerca, autonoma e attraverso la partecipazione a progetti nazionali ed internazionali, sui seguenti argomenti:

- progettazione e sviluppo di algoritmi paralleli portabili per il riconoscimento di forme ed il trattamento di dati spaziali con diverse applicazioni;
- progettazione e sviluppo di algoritmi paralleli per il trattamento dell'incertezza nell'analisi di dati spaziali e per il riconoscimento di forme;
- studio, progettazione e implementazione di metodi e strumenti per lo sviluppo di applicazioni su piattaforme di calcolo eterogenee.

Ferrania Imaging Technologies è una multinazionale italiana, presente con otto sedi in sei paesi ed export nel resto del mondo, con sede legale in provincia di Savona.

L'oggetto principale dell'attività è soprattutto la produzione di materiali fotosensibili e per il trattamento di immagini e le sue evoluzioni nel mondo digitale. Ferrania è presente nel settore del trattamento delle immagini da oltre ottanta anni e oltre gli aspetti di produzione, svolge un'intensa attività di ricerca e sviluppo. La produzione è concentrata negli stabilimenti di Ferrania - (SV) Italia e Weatherford, Oklahoma.

Il settore Ricerca e Sviluppo è situato in Ferrania (SV). Il settore Ricerca e Sviluppo è leader nell'innovazione per quanto riguarda le tecnologie foto sensibili, ha un ampio staff di tecnici e ricercatori in diverse discipline e possiede un considerevole portafoglio di brevetti.

Ferrania ha inoltre una propria struttura di Information e Communication Technology recentemente ridisegnata e basata su soluzioni innovative che collegano la rete aziendale locale e le diverse sedi geografiche.

7.4. Funzionalità e strumenti

La libreria in questione fornirà le seguenti funzionalità:

- Acquisizione di immagini di natura eterogenea (da fonti differenti ed in formati differenti) e memorizzazione delle stesse in una base dati dotata di uno schema comune (eventualmente composta da n data base di natura eterogenea)
- Algoritmi per il trattamento e l'analisi delle immagini
- Algoritmi di compressione delle immagini
- Algoritmi di conversione del formato delle immagini
- Algoritmi di ricerca delle immagini (totali o parziali) nella base dati

8. Task 5: Applicazione in ambito di controllo ambientale e protezione civile

L'applicazione finale del progetto è stata individuata nel settore del *controllo ambientale e protezione civile*, allo scopo di sviluppare e sperimentare sistemi integrati ad alte prestazioni e in tempo reale per il telerilevamento, considerato l'interesse che questi hanno a livello di enti locali e società di servizio operanti sul territorio.

In particolare la presente proposta considera come ambito applicativo il *processamento SAR interferometrico* nei suoi due principali aspetti: quello relativo al processamento dei dati SAR per la generazione dei DEM (Digital Elevation Model), cioè alla mappatura delle altezze, e quello dell'interferometria differenziale, che consente di misurare, anche su intervalli di tempo abbastanza lunghi, piccoli movimenti del terreno con accuratezze dell'ordine del centimetro. Questa attività è volta a consentire il monitoraggio e la gestione dei rischi naturali.

Si è preso in considerazione un intero ambito applicativo piuttosto che una singola applicazione, nell'intento di migliorare le prestazioni dello scenario operativo: nella prima fase del progetto saranno focalizzati i passi di processamento maggiormente critici e che meglio si prestano all'ottimizzazione parallela.

L'attività coordinata dall'ASI comprende sia lo sviluppo di ambienti specializzati di supporto alle applicazioni (livello L3), sia lo sviluppo di applicazioni (livello L4).

8.1. Integrazione di librerie per elaborazione di immagini telerilevate

Gruppi: ASI (Milillo), IESI (Veneziani, Lovergine), PoliBa (Guerriero)

Nell'ambito dell'Elaborazione di immagini, l'attività dell'area sarà indirizzata all'applicazione del modello del livello L2 al contesto della *elaborazione delle immagini satellitari (Telerilevamento)*.

L'impostazione del progetto nasce dalla constatazione che la massima parte della ricerca e dei prodotti applicativi nell'area del Telerilevamento si basa su strumenti software commerciali avanzati di alto livello, che rappresentano un riferimento ed uno standard de facto. Strumenti di base importanti sono rappresentati da ambienti software consolidati, come Matlab (Mathworks Inc.), IDL (RS, Inc.) e PV-Wave (Visual Numerics, Inc.), EarthView (Atlantis Sc. Inc.), oppure da tools di analisi interattiva sviluppati sulla base di questi, come Envi (RS, Inc.) ed ER-Mapper/ER-Radar (ERM Pty Ltd.), o ancora più specializzati, come nel caso ancora di EV-InSAR (Atlantis Sc. Inc.), dedicato all'interferometria SAR (synthetic aperture radar). Ciascun ambiente è dotato di linguaggio di controllo dedicato interpretato, ed include ampie librerie di comandi macro, attraverso cui vengono effettuate anche elaborazioni di complessità superiore. Nel caso di Matlab, poi, il prodotto (multipiattaforma) dispone di un cross-compiler attraverso cui è possibile realizzare librerie di moduli oggetto o produrre eseguibili nel linguaggio macchina della piattaforma target, generati a partire dai sorgenti in linguaggio di comando, opportunamente pre-processati e tradotti in un linguaggio intermedio convenzionale (linguaggio C). Tutti questi ambienti rendono immediatamente disponibili, quindi, un ricco set di librerie e costrutti di alto livello, coordinati da un linguaggio di per sé stesso di un livello di astrazione maggiore rispetto ai linguaggi convenzionali di basso livello come C, Fortran o Java, che ad esempio integrano a livello sintattico il calcolo matriciale o simbolico, con operatori e costrutti elementari. Questo tipo di linguaggio consente, nello sviluppo di applicazioni, un alto livello di riusabilità del codice, astrazione, sintesi e rapida prototipizzazione. Non si può di conseguenza prescindere dall'integrare questo tipo di strumenti in un ambiente di programmazione del livello L2, possibilmente con lo stesso grado di flessibilità e modularità.

8.2. Catena di processamento SAR interferometrico

Gruppi: ASI (Milillo), Telespazio (Lopinto, Candela), ISUFI (Alosio), UniBa2 (Di Lecce)

Considerando l'approccio generale di questo progetto, l'attività proposta si posiziona al livello *L4 - Applicazioni*, e si pone come obiettivo la sperimentazione su piattaforma parallela ad alte prestazioni di un'applicazione, complessa ed articolata, nell'ambito delle Osservazioni della Terra: *il processamento interferometrico dei dati acquisiti dallo Shuttle nell'ambito della Terza missione SRTM ed esteso anche ai dati ERS*. Si tratta di una catena di processamento già operativa sviluppata dalla Telespazio nell'ambito di un contratto ASI, che elabora i dati acquisiti durante la missione SRTM per ottenere come prodotto finale i DEM (Digital Elevation Model) interferometrici [32]. I passi di processamento presenti nella catena SRTM sono:

- ✓ XSAR-SRTM Image processing
 - Processore SSC
 - Processore MGD
 - Processore SSC Interferometrico

- ✓ AODA data Processing
- ✓ Electrical Phase Calibration
- ✓ Interferogram and coherence generation and filtering
- ✓ Phase unwrapping
- ✓ Image geometric calibration
- ✓ Geocoding and Mosaiking
- ✓ Image quality analysis and validation

Accanto ai sottosistemi illustrati in tale articolo, si considerano anche altri due elementi di processamento:

- ✓ un simulatore di dati SAR (SSC e fase interferometrica)
- ✓ un processore per interferometria differenziale, DIFSAR, basato sui Permanent Scatters (si rimanda all'allegata descrizione per i dettagli).

Nelle fasi preliminari del progetto verrà svolta un'accurata analisi di tutti i processori presenti nella catena, al fine di individuare quelli che – per onerosità e criticità – trarrebbero beneficio da un'implementazione parallela ad alte prestazioni: saranno questi ultimi ad essere oggetto della sperimentazione successiva.

9. Indicatori per la valutazione del raggiungimento degli obiettivi di progetto

Oltre ai classici indicatori legati alla produzione scientifica nel contesto scientifico internazionale, la valutazione dei risultati del progetto è legata principalmente al **prototipo** di piattaforma, alla sua documentazione e ed alla sua divulgazione nei confronti della comunità scientifica, industriale e degli utenti.

La documentazione, disponibile via pagina Web, consisterà in manuali utente ed in rapporti tecnico-scientifici relativi a tutte le parti del prototipo.

Verrà costituito un comitato di utenti che si coordinerà con la istituenda *Associazione Italiana per il Calcolo ad Alte Prestazioni*, che includerà aziende di informatica e di calcolo avanzato (costruttori, independent software vendors), utenti, mondo della ricerca rappresentato dal Progetto.

Nei riguardi dell'applicazione (Task 5), ASI effettuerà valutazioni con la Protezione Civile, Ambiente, eventualmente Difesa ed altri Ministeri interessati, nonché industrie e partner europei.

Verranno effettuati due workshop, rispettivamente alla fine del primo e del secondo anno.

10. Articolazione temporale del progetto

Il progetto ha la durata di 2 anni.

Primo anno:

- *Task 1:* specifica e realizzazione della versione α dell'ambiente.
- *Task 2, 3, 4:* specifica e sperimentazione delle componenti specializzate utilizzando codici sequenziali disponibili e strumenti esistenti che consentano di emulare la metodologia dell'ambiente del progetto.
- *Task 5:* individuazione delle parti critiche della catena di processamento interferometrico dell'ASI, studio della parallelizzazione con strumenti esistenti che consentano di emulare la metodologia dell'ambiente del progetto.

Secondo anno:

- *Task 1:* realizzazione della versione β dell'ambiente.
- *Task 2, 3, 4:* porting delle componenti specializzate utilizzando, per la parallelizzazione, la versione α dell'ambiente. Integrazione delle librerie nell'ambiente. Porting finale con la versione β dell'ambiente.
- *Task 5:* parallelizzazione delle parti critiche utilizzando la versione α dell'ambiente. Porting finale con la versione β dell'ambiente.

11. Riferimenti

1. K. Keahey, P. Beckman, J. Ahrens, "Ligature: component architecture for high performance applications", *The International Journal of High Performance Computing Applications*, vol. 14, n. 4, Winter 2000, 347-356.
2. K. Keahey, D. Gannon, "PARDIS. a parallel approach to CORBA", *6th IEEE Int. Symp. on High Performance Distributed Computing*, 1997, 31-39.
3. R.C. Armstrong, A. Chung, "POET (parallel object-oriented environment and toolkit) and frameworks for scientific distributed computing", *Hawaii Int. Conf. on System Sciences*, 1997.
4. R. Armstrong, D. Gannon, et al, "Toward a common component architecture for high-performance scientific computing", *8th IEEE Int. Symp. on High Performance Distributed Computing*, 1999.
5. *Common Component Architecture Forum*, www.acl.lanl.gov/cca-forum.
6. B. Smolinski, S. Kohn, N. Elliott, N. Dykman, "Language interoperability for high-performance parallel scientific components", *LNCS 1732*, 1999, 61-71.
7. D. Goswami, A. Singh, B. Preiss, "Using object-oriented techniques for realizing parallel architectural skeletons", *LNCS 1732*, 1999, 130-141.
8. Object Management Group, *The Common Object Request Broker: architecture and specification*, 1995.
9. I. Foster, C. Kesselman, *The Grid: blueprint for a new computing infrastructure*, Morgan Kaufmann, 1999.
10. Proposta di Progetto CNR, sottoprogetto *Grid Computing: Tecnologie Abilitanti e Applicazioni per eScience*, Fondo Speciale 1999.
11. Hey et al, "Integrating computation and information resources: an MPP perspective", in *Massively Parallel Programming Models*, IEEE Computer Society, 1998.
12. M. Vanneschi, "Heterogeneous HPC environments", invited paper, *4th Int. Euro-Par Conference*, Southampton, Sept. 1998, in D. Pritchard and J. Reeve (Eds.), *Lecture Notes in Computer Science*, vol. 1470, pp. 21-34.
13. M. Vanneschi, "PQE2000 : HPC tools for industrial applications". *IEEE Concurrency*, IEEE Computer Society, October-December 1998, pp. 68 – 73.
14. Gruppo di Lavoro MURST, *Proposta di PNR sul Calcolo ad Alte Prestazioni*, luglio 1997
15. J.J. Dongarra, D.W. Walker, "Software Libraries for Linear Algebra Computations on High Performance Computers", *SIAM Review*, vol. 37, 1995.
16. D. Di Serafino, L. Maddalena, P. Messina and A. Murli, "Some perspectives on High Performance Mathematical Software", in *High Performance Algorithms and Software in Nonlinear Optimization*, R. De Leone, A. Murli, P. Pardalos and G. Toraldo Editors, Kluwer Academic Press, 1998.
17. G. Aloisio., N. Veneziani., G.C. Fox, G. Milillo, "Computational load evaluation for the real-time compression of X-SAR raw data", *Space Technology*, Apr. 1990.
18. R. Albrizio, A. Mazzone, N. Veneziani, G. Aloisio and P. Messina, "Performance of multiprocessor structures for fast digital SAR-processing". In Q. Stout and M. Wolfe Eds, *The Sixth Distributed Memory Computing Conference Proceedings*, pp. 611-616, IEEE Computer Society Press, 1991.
19. R.G. Healey (ed), Special issue on parallel processing in GIS, *Int. Journal of GIS*, Oct. 1996.
20. A. Clematis, A. Coda, B. Falcidieno, M. Spagnuolo, "High-performance computing for surface modelling and analysis", *The Visual Computer*, vol. 16, n. 1, pp. 62-78, January 2000.
21. S.A. Zenios, "High-Performance Computing in finance: the last 10 years and the next", *Parallel Computing*, Dec. 1999.
22. *Workshop on Large-Scale Parallel KDD Systems*, Aug. 1999, in conjunction with 5th ACM SIGKDD Int. Conf. on KDD 99.
23. Proposta di Progetto CNR, sottoprogetto *Tecnologie per arricchire e fornire accesso a contenuti - linea Mine the WEB*, Fondo Speciale 1999.
24. P. Ferragina, F. Luccio, "String search in coarse-grained parallel computers", *Algorithmica: Special issue on Coarse-Grained Parallel Computers*, Editor Frank Dehne, vol. 24, 177--194, 1999.
25. P. Ferragina, G. Mancini, "Opportunistic data structures with applications", *IEEE Foundations of Computer Science (FOCS)*, 2000.

26. P. Ferragina, G. Mancini, "An experimental study of an opportunistic index", *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2001.
27. L. A. Freitag, W. D. Gropp, P. D. Hovland, L. C. McInnes, B. F. Smith, "Infrastructure and Interfaces for Large-Scale Numerical Software", in *Proc. of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications*, June 28 - July 1, 1999, Las Vegas, Nevada.
28. S. Filippone, M. Colajanni, "PSBLAS: a Library for Parallel Linear Algebra on Sparse matrices", *ACM Transactions on Mathematical Software*, 26(4), 2000.
29. M.R. Guarracino, F. Perla, "A Parallel Block Lanczos Algorithm for Distributed Memory Architectures", *Journal of Parallel Algorithms and Applications*, 4, 1995.
30. Antonio Laganà, "Innovative Computing and Detailed Properties of Elementary Reactions Using Time Independent Approaches", *Computer Physics Communications*, 116, 1999.
31. T. Sommerfeld, F. Tarantelli, "Subspace iteration techniques for the calculation of resonances using complex symmetric Hamiltonians", *Journal of Chemical Physics* 112, 2000.
32. R. Loizzo, C. Manzo, L. Antonucci, L. Candela, M. Costantini, P. Inversi, E. Lopinto, G. Riccobono, N. Santantonio, A. Spinelli, G.R. Verdone, R. Viggiano, "I-PAF SRTM X-SAR Processing Chain : Algorithms and Results", *IGARSS 2000*.