

6° CONGRESSO NAZIONALE CIRIAF

SUITE INTEGRATA DI GESTIONE DI MODELLI DI CALCOLO PER L'IMPATTO AMBIENTALE DI RUMORE, ATMOSFERA E CAMPI ELETTROMAGNETICI.

Prof. Angelo Farina¹, Ing. Paolo Galaverna², Dott. Guido Truffelli³

¹Università degli Studi di Parma

²Genesis Acoustic Workshop, Parma,

³Laboratorio di Acustica ed Elettroacustica, Parma

SOMMARIO

Il lavoro svolto ha avuto come scopo quello di creare una piattaforma software in grado di accogliere al suo interno l'analisi di più componenti ambientali per avere una visione complessiva di un intervento, nonché un'ottimizzazione dei tempi e dei dati da acquisire ed inserire nel motore di calcolo. Il sistema modellistico integrato nasce dall'esperienza di vari progetti sviluppati nel passato: il software Ramsete e il progetto DISIA per il rumore, il modello MoniQA (Progetto LIFE-94) per l'atmosfera e di uno riguardante i campi elettromagnetici in fase di realizzazione presso il Dipartimento di Ingegneria dell'università di Parma; l'unione delle esperienze e dei vari gruppi di lavoro ha portato a sviluppare un nuovo strumento software che si pone sul mercato integrando alcuni classici strumenti di modellistica in una struttura di ultima generazione.

Nasce così un programma per l'analisi dell'impatto di sorgenti di inquinanti di svariati tipi: acustico, atmosferico o elettromagnetico. Vengono utilizzati alcuni dei modelli di calcolo più diffusi (RLS, Shall, ISC3, CALINE, ...) ma il sistema è aperto all'integrazione di nuove procedure di calcolo partendo da un modello uniformato dell'ambiente oggetto di studio.

1.. LA BASE SOFTWARE ESISTENTE

Uno dei primi progetti realizzati dal Laboratorio di Acustica ed Elettroacustica (LAE) è stata la riscrittura del software per il calcolo dell'impatto acustico di strade e ferrovie denominato CityMap con un duplice obiettivo: raggiungere un significativo incremento delle prestazioni; recepimento delle nuove normative Europee nel campo della simulazione. Il primo obiettivo è stato pienamente raggiunto, riscrivendo il programma in linguaggio C++ e ricorrendo ad algoritmi calcolo ottimizzati per il problema specifico. Tale software soffriva tuttavia di alcune limitazioni: pur permettendo di associare le proprietà di emissione ad ogni sorgente lineare o puntiforme in maniera molto dettagliata, non era dotato di nessuno strumento di inserimento o modifica del modello geometrico, per cui l'utente era obbligato a servirsi di uno o più programmi esterni, che fossero in grado di esportare le più semplici primitive geometriche (punti, cerchi, poligoni) in formato DXF, cioè il formato standard di interscambio dati di Autocad. Questa limitazione nell'editing delle geometrie era ed è tuttora comune a moltissimi software commerciali e non, prodotti all'interno di enti di ricerca o di strutture di consulenza pubbliche e private. La maggior si basa sulla importazione da Autocad, ArcInfo o altri software di terze parti per l'importazione del modello dell'ambiente e delle fonti di inquinamento esistenti, mentre gestisce internamente le fonti simulate. Alcuni si basano su file di testo in formato proprietario inseriti manualmente o quasi. Durante lo sviluppo del programma di simulazione acustica per interni Ramsete, nel corso degli anni è stato realizzato un notevole insieme di strumenti piuttosto sofisticati ed al tempo stesso molto semplici per qualsiasi utilizzatore, che permettono l'inserimento del modello geometrico su cui effettuare le simulazioni. L'ultima versione di tale software contiene alcune estensioni, pensate per il suo utilizzo anche in ambiente

esterno, ad esempio: si può facilmente importare un'immagine vettoriale della cartografia e selezionare in maniera veloce le piante degli edifici e generare il loro modello tridimensionale.

2.. LA NUOVA SUITE

Al momento sono in fase di implementazione o reimplementazione ben tre software per la simulazione della propagazione del rumore in ambiente esterno, del inquinamento atmosferico ed elettromagnetico. Vista la base esistente ed osservando la grande quantità di esigenze comuni a tutti i modelli di calcolo per le diverse componenti ambientali, si è pensato di realizzare un'unica suite integrata per la gestione di tutte le simulazioni, denominata appunto RamSuite.

Osservando la sequenza dei passi necessari a compiere una qualsiasi simulazione, è praticamente immediato riconoscere alcuni elementi comuni indipendenti dallo specifico problema da analizzare:

1. Inserimento del modello del terreno
2. Inserimento degli edifici ed altri manufatti
3. Inserimento delle sorgenti
4. Caratterizzazione delle sorgenti
5. Definizione della griglia dei ricettori
6. Esecuzione del calcolo
7. Visualizzazione dei risultati

Praticamente, si nota che gli unici punti, che differenziano sostanzialmente una simulazione dall'altra, sono il 4 ed il 6, cioè la definizione dei parametri specifici delle sorgenti ed il processo di calcolo vero e proprio. Visto che tutte le altre fasi sono praticamente identiche, è davvero insensato inventarsi strumenti ad hoc per ogni nuovo modello, molto meglio sarebbe utilizzare sempre la stessa interfaccia utente e produrre un file in un unico formato predefinito.

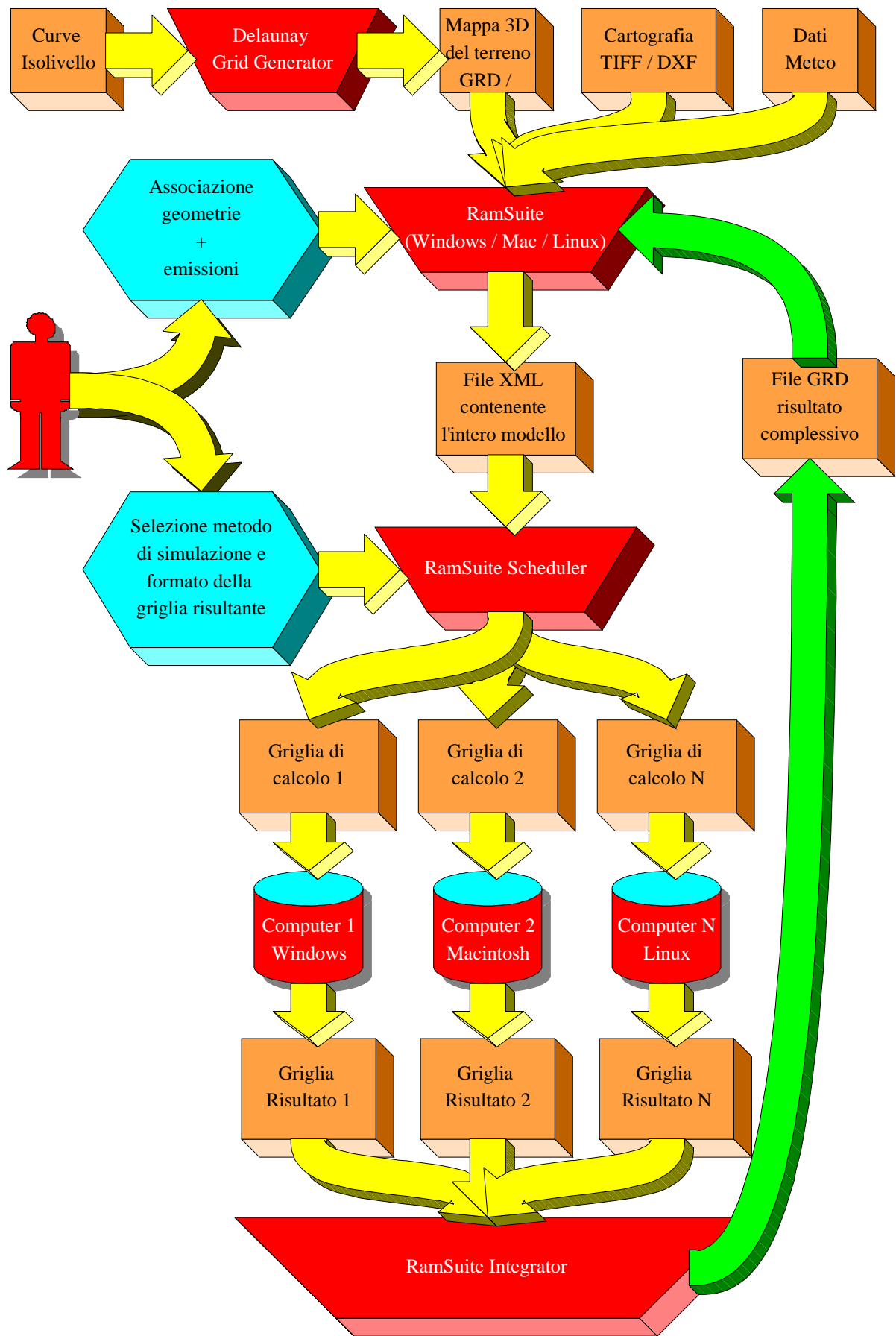


Figura 1. Flusso dei dati in una sessione tipica di lavoro con RamSuite

In figura 1 è rappresentato il flusso dei dati di una tipica sessione di lavoro con RamSuite. I rettangoli color arancio rappresentano i file che di volta in volta vengono elaborati, i trapezi rossi rappresentano invece le fasi di elaborazione. Le frecce gialle mostrano il percorso dei dati di ingresso, mentre quelle verdi rappresentano il percorso dei dati in uscita. Di seguito vediamo i vari passi e le soluzioni adottate.

2.1. Inserimento del modello del terreno

Quando le dimensioni della zona da studiare superano quelle del piccolo quartiere, non è possibile accontentarsi dell'assunzione di operare su di un terreno perfettamente piano, posto convenzionalmente a quota zero. Di conseguenza è necessario procurarsi una rappresentazione sufficientemente precisa dell'altimetria del terreno. I programmi calcolo solitamente ricevono in ingresso delle mesh tridimensionali di triangoli o quadrilateri piani, purtroppo tale modello non è quasi mai disponibile oppure lo è a costi proibitivi. Molto spesso invece è possibile procurarsi o ricavare da un disegno le curve isolivello, per cui il problema si traduce semplicemente nella trasformazione di queste curve in una maglia di triangoli, praticamente un file DXF contenente esclusivamente entità di tipo 3DFACE, oppure griglia regolare nel formato GRD riconosciuto anche da Surfer. Il modulo della suite, preposto a questo compito, attualmente legge esclusivamente file DXF, da cui estrae le entità di tipo POINT, LINE, POLYLINE, LWPOLYLINE e 3DFACE. Partendo da queste primitive genera una griglia uniforme di triangoli, la quale può essere infittita nei punti di maggiore interesse o in cui il terreno presenta asperità più marcate.

E' ancora in corso di analisi anche la procedura di generazione di una griglia tridimensionale non uniforme, al fine di ottenere una maglia di triangoli di De Launay¹ a partire dai nodi delle polilinee in ingresso.

2.2. Inserimento di sorgenti e barriere

Una volta importato il modello del terreno è necessario inserire le sorgenti, cioè tutte le fonti di "inquinamento" che si devono studiare. Nel campo del rumore le sorgenti sono rappresentate da strade, ferrovie o generiche sorgenti puntiformi.



Figura 2. Importazione del modello geometrico

Nel campo dell'inquinamento atmosferico si devono individuare strade, ferrovie (diesel), fabbriche, abitazioni, inceneritori. Nel campo dell'inquinamento elettromagnetico si devono definire le linee ad alta tensione, le ferrovie

(elettriche), le centrali di produzione. In tutti i casi, praticamente il lavoro si traduce nell'individuare le linee, i punti e le aree che devono essere considerate come sorgenti dal programma di calcolo. Ovviamente è possibile modificare il modello, ad esempio per aggiungere nuove strade o sorgenti non ancora esistenti al fine di prevederne l'impatto. Spesso è anche necessario anche semplificare il dettaglio di alcune entità al fine di ottimizzare i tempi di calcolo, che possono dipendere direttamente dal numero di nodi delle polilinee. Ad esempio è normalmente inutile approssimare le curve di una strada con decine di segmenti di 1 metro, soprattutto quando le abitazioni distano centinaia di metri da questa curva. Per questo motivo è possibile aggiungere ma soprattutto eliminare nodi da qualsiasi polilinea importata.

2.3. Caratterizzazione delle sorgenti

Dopo l'inserimento delle caratteristiche geometriche, si devono assegnare tutti i parametri di emissione delle varie fonti. Tali parametri sono molto diversi a seconda del tipo di sorgente e spesso variano anche col tipo di algoritmo di simulazione. Ad esempio, quando si deve caratterizzare il traffico stradale, alcuni programmi classificano tutti i veicoli in tre sole categorie, altri ne riconoscono almeno sei, altri ignorano completamente i ciclomotori. Per questo motivo non è stato possibile costruire un'interfaccia comune per l'inserimento di tali attributi, ma si sono dovute realizzare tante maschere di inserimento specifiche caso per caso, in modo da poter proporre valori standard per tutti i parametri ed eseguire appropriati controlli contestuali sulla validità dei valori inseriti.

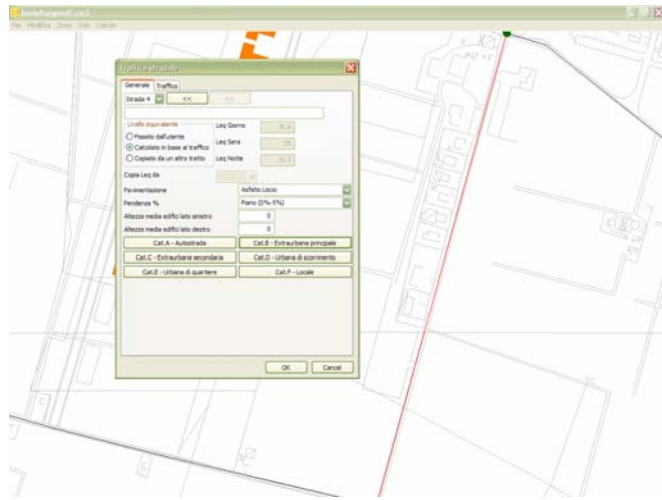


Figura 3. Definizione dei parametri della strada selezionata

In figura 3 è possibile vedere la maschera per la caratterizzazione di una strada, dove è possibile inserire i dati sul traffico suddivisi per categorie di veicoli, oppure è possibile selezionare un profilo standard da una lista predefinita di tipologie di strada: autostrade, extraurbane, urbane, locali, ecc.

2.4. Definizione della griglia dei ricettori

L'ultima fase precedente il calcolo consiste nella selezione dei ricettori, ossia dei punti chiave in cui verranno misurati gli effetti delle vari sorgenti. In figura 4 vediamo la semplice maschera di definizione di una griglia di ricettori disposti ad intervalli regolari su di piano ad una quota prefissata sul terreno.

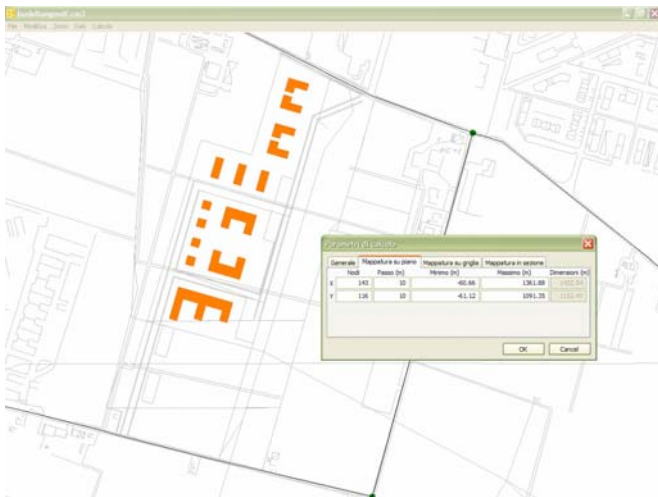


Figura 4. Definizione della griglia dei ricettori

Ovviamente è possibile definire anche griglie verticali od orientate a piacere. Al limite è possibile specificare un elenco di punti critici ben precisi sia importando le entità POINT da un file DXF oppure inserendole manualmente. Nella maggior parte dei casi, il numero di ricettori influenza pesantemente i tempi di calcolo, per cui è utile ridurre al massimo il loro numero nei punti lontani dalla zona d'interesse e definirne un numero congruo nei punti vicini. A tale scopo è stato approntato uno strumento di generazione automatica della lista dei ricettori il quale, partendo da una griglia iniziale, è in grado di eliminare i punti troppo lontani dalle sorgenti, dalle barriere o dalle abitazioni. Nel risultato finale, tutti i punti eliminati verranno poi calcolati interpolando i valori noti circostanti

2.4. Esecuzione del calcolo

Una volta completata la definizione di tutti i parametri del modello per la elaborazione è necessario specificare il metodo di calcolo desiderato e di conseguenza i parametri specifici del metodo prescelto. Tali parametri possono essere molto diversi a seconda del problema, per cui anche in questo caso non è stato possibile utilizzare una maschera universale per il loro inserimento. Di conseguenza, per ogni singolo motore di calcolo è stata definita una diversa maschera per l'inserimento dei parametri, che possono spaziare dalle condizioni meteo reali o simulate alla data e durata dell'intervallo di tempo in cui avviene la simulazione. Il programma principale della suite a questo punto salva un file di testo in formato XML contenente tutte le informazioni sul modello e poi, su indicazione dell'utente, esegue il calcolo vero e proprio localmente oppure è in grado di distribuire il lavoro su più computer sulla rete, nel modo descritto nella sezione 4.

2.5. Visualizzazione dei risultati

La visualizzazione dei risultati di un'elaborazione è un compito solitamente molto complesso, in quanto il volume dei dati prodotti è piuttosto elevato, per cui si devono prevedere svariate modalità di visualizzazione in forma di tabelle, grafici o mappe. Ognuna di queste rappresentazioni è strettamente legata al problema specifico e per il momento si è quindi preferito limitarsi esportare i risultati in un formato facilmente elaborabile con strumenti prodotti da terze parti. Storicamente per i tutti i programmi da noi prodotti finora, hanno sempre eletto Surfer ed Excel a programmi preferiti per lo studio e l'elaborazione grafica dei risultati.

Tra tutti i risultati di un'elaborazione tuttavia è possibile definire quale sia il dato principale, che possa essere mappato in trasparenza sul modello geometrico iniziale in modo da avere un immediato riscontro visivo dell'esito della elaborazione. Ad esempio, come si vede in fig.4, nel caso di CityMap 3.0 il risultato della elaborazione può essere riassunto nella mappa del livello equivalente diurno (Leq_{day}).

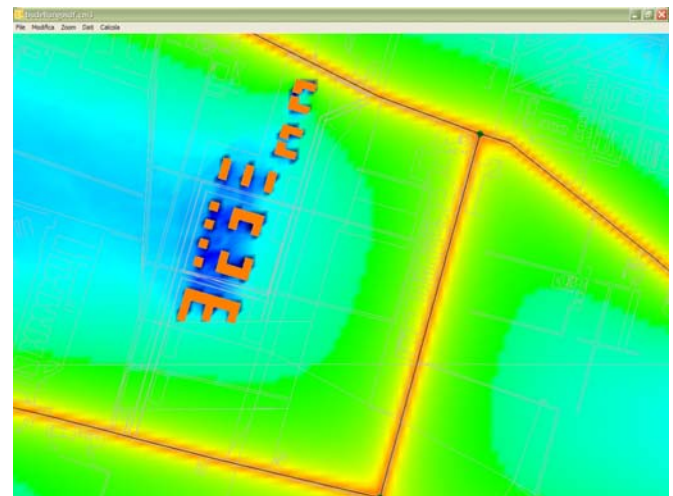


Figura 5. Mappatura del Leq_{day}

3. LA "LEGGE" DI MOORE

La cosiddetta legge di Moore, enunciata da uno dei fondatori di Intel nel 1964, affermava che "la velocità dei microprocessori raddoppia ogni 18 mesi"

L'osservazione era puramente empirica e, stando a Moore stesso, non pretendeva di diventare una vera e propria regola; non pareva possibile che la corsa al raddoppio potesse continuare indefinitamente, tuttavia si prevedeva che sarebbe rimasta valida almeno fino al 2017. Tuttavia, come ogni utilizzatore abituale di sistemi informatici avrà certamente notato, negli ultimi anni l'accelerazione dei microprocessori non ha affatto rispettato queste aspettative: attualmente la velocità raddoppia almeno ogni 30 mesi. In realtà, non è affatto semplice definire il concetto di velocità di un computer nel suo complesso, in quanto i fattori che ne determinano le prestazioni non sono limitati al solo processore centrale, ma comprendono anche tutte le periferiche ad esso collegate: vari livelli di cache interne ed esterne, RAM, Hard disk, scheda video, scheda ethernet, ecc. Se quindi non ci limitiamo a considerare semplicemente i GHz dichiarati dal produttore del singolo processore, ma valutiamo le prestazioni complessive di un sistema di ultima generazione, che si avvalga di tecnologie come l'hyperthreading e/o di processori dual-core o in generale multi-core, notiamo che la legge di Moore torna ad essere parzialmente verificata.

3.1 Algoritmi per il calcolo distribuito

Fino a qualche tempo fa il software ha potuto godere dell'aumento di prestazioni a costo praticamente nullo, limitandosi a sfruttare semplicemente l'aumento della velocità del singolo processore centrale. Al momento questa epoca d'oro sembra essere finita: per poter aumentare le prestazioni, il software deve essere riprogettato in modo tale da sfruttare attivamente l'eventuale presenza di più unità di calcolo in parallelo. Lo sforzo necessario per lo sviluppo delle cosiddette applicazioni multi-thread è ordini di grandezza più elevato rispetto a quello delle applicazioni single threaded. Si devono utilizzare sistemi di condivisione

delle risorse per evitare collisioni e deadlock, il tutto senza l'ausilio dei tradizionali strumenti di individuazione e correzione degli errori, resi quasi inutilizzabili quando condizioni causanti un errore sono difficilmente ripetibili. Per ottenere il massimo delle prestazioni è allora necessario utilizzare i compilatori realizzati dagli stessi produttori del processore oppure, scendendo ancora a più basso livello, ricorrere direttamente al linguaggio macchina, scelta che storicamente non ha mai ripagato, vista la sfrenata corsa all'innovazione in questo campo.

Tuttavia, nel caso specifico dei modelli di calcolo attualmente supportati dalla suite, in fondo non è strettamente necessario riscrivere tutto in modo da sfruttare il parallelismo nel calcolo. Di fatto, tutti i singoli programmi di calcolo possono essere eseguiti già ora in parallelo: basta separare la griglia dei ricettori completa in varie porzioni e quindi dare in pasto a più copie dello stesso programma di calcolo, sia sulla stessa macchina dual-core oppure su macchine diverse. Alla fine delle elaborazioni basta raccogliere e combinare opportunamente tali risultati, di norma sommabili direttamente.

Questo processo, se eseguito manualmente, oltre ad essere estremamente tedioso, ovviamente comporta un notevole grado di inaffidabilità, inoltre la distribuzione del carico potrebbe risultare sbilanciata causando notevoli fastidi agli altri utenti del sistema. Si pensi che una recente statistica afferma che normalmente viene utilizzato solo il 5% della potenza di calcolo dei computer all'interno di una rete locale. Gli utenti che scrivono testi o elaborano fogli di calcolo, non richiedono che una minima parte della potenza di calcolo rispetto a quelli che eseguono simulazioni numeriche. Come se non bastasse, di fatto molti computer passano intere ore ad eseguire inutili screen savers. In molti si sono accorti di questo inutile spreco e per sfruttare al massimo questa potenza, sono stati sviluppati molti sistemi per l'elaborazione distribuita o grid computing, come ad esempio il Globus Toolkit oppure il progetto BOINC² dell'università di Berkeley, che distribuisce addirittura su scala mondiale processi di calcolo per gli ambiti più svariati: la ricerca di numeri primi; lo studio delle proteine e del DNA; l'analisi dei segnali provenienti dallo spazio ([Seti@home](#)). Questo sistema è basato proprio su di uno screen saver che, quando il computer è inattivo, elabora i dati ricevuti da un server centrale.

4. IL PROTOCOLLO DI RAMSUITE

Tutti questi sistemi, basati solitamente su web services, a nostro parere richiedono un'infrastruttura di rete non indifferente e conseguentemente difficile da amministrare, se non in presenza dei cosiddetti guru del sistema.

Abbiamo quindi pensato di realizzare un ambiente il più semplice possibile, in modo che potesse funzionare normalmente su una singola macchina ed opzionalmente potesse utilizzare anche altri computer, all'interno della stessa rete su cui sia stata installata RamSuite. Il tutto senza dover ricorrere ad un server centralizzato e magari sfruttando macchine con sistemi operativi diversi, che altrimenti rimarrebbero inutilizzate.

Il programma principale, una volta generato il file XML col modello geometrico e quello della griglia dei ricettori sarebbe pronto all'esecuzione dell'elaborazione. L'utente potrà decidere semplicemente se eseguire l'intero processo in locale, oppure se tentare di sfruttare anche altri computer in rete. In questo secondo caso partirà un modulo che si occuperà delle seguenti operazioni:

1. generare la lista dei computer disponibili (slave)
2. assegnare ad ogni slave un task (modello+ricettori)
3. attendere la fine delle elaborazioni
4. riassegnare i task terminati inaspettatamente
5. integrare i risultati parziali nel risultato complessivo

4.1. La lista degli slave

Al momento dell'installazione sarà possibile decidere se installare un programma detto slave, il quale rimarrà perennemente in attesa su di una porta TCP predefinita, della richiesta di un'elaborazione. La priorità di tale processo sarà minima in modo da non interferire minimamente con l'uso normale del sistema. Il programma principale della suite, al momento del lancio della simulazione, interroga tutti gli slave disoccupati in modo da stabilire una ottimale suddivisione del carico.

4.2. Distribuzione del lavoro

Una volta determinato il numero di slave, il programma non deve fare altro che suddividere equamente tra loro la lista completa dei ricettori ed inviare ad ognuno sia il modello, sia la lista tramite un protocollo SOAP/HTTP. Per ragioni di sicurezza, si è deciso di non inviare anche l'eseguibile stesso per l'elaborazione.

4.3. Attesa dei risultati

Durante l'attesa dei risultati, il programma principale deve interrogare periodicamente gli slave, in modo da verificare lo stato di avanzamento dei lavori.

4.4. Riassegnazione di un task

Può succedere che la connessione con uno slave venga disattivata per un motivo qualsiasi: il computer si è bloccato inaspettatamente, è stato spento involontariamente, la rete è scollegata, ecc. Non appena questa situazione viene rilevata, il programma principale deve riattribuire il lavoro interrotto ad un altro slave disponibile, eventualmente quello in esecuzione sul computer stesso che ha lanciato il programma principale.

4.5. Somma dei risultati

Quando tutti gli slave hanno risposto (sempre via TCP) i loro risultati, il programma può sommarli tra loro ed avvertire l'utente del completamento del lavoro.

5. PIATTAFORME SUPPORTATE

Ogni volta che si inizia un nuovo progetto, si innanzi tutto definire bene quale sia il tipo di utenza cui il prodotto dovrebbe rivolgersi. Una volta stabilita la classe di utenza, spesso si determina quasi di conseguenza anche quale sia la piattaforma hardware e software tipica di questi utenti. Fino ad ora però avevamo sempre prodotto eseguibili per le varie versioni di Windows. La prima versione del software Ramsete a 16 bit era già compatibile con Microsoft Windows 3.0 e 3.1 for Workgroups. Successivamente è stato portato in ambiente a 32 bit (da Win 95 a Win XP) e certamente passerà anche ai 64 bit. Tuttavia il nuovo ambiente RamSuite si rivolge sia ai professionisti sia agli ambienti di ricerca e controllo dell'ambiente privati e pubblici. Ultimamente gran parte delle linee guida, spinge all'uso di software e sistemi operativi non proprietari od open-source. Dopo un lungo periodo di stasi, anche i sistemi Macintosh stanno conquistandosi una percentuale crescente di un mercato che prima era confinato alla sola nicchia dell'editoria. In risposta a questi segnali abbiamo realizzato

la nuova suite in modo che possa essere facilmente distribuita non solo in ambiente Windows ma anche Linux e Macintosh.

In un primo momento è stato valutato l'utilizzo del nuovissimo linguaggio C# il quale, pur essendo stato realizzato da Microsoft, è stato successivamente portato su tutte le altre piattaforme da un progetto open-source denominato Mono. Tuttavia il sistema non è ancora del tutto maturo e non sono disponibili librerie standard per la realizzazione di interfacce grafiche per tutte le piattaforme. Alla fine abbiamo optato per il più tradizionale linguaggio C++, combinato con le librerie open-source denominate wxWidgets. Tali librerie sono completamente gratuite e possono essere utilizzate con qualsiasi compilatore C++ su decine di piattaforme. Hanno raggiunto un notevole livello di affidabilità e stabilità, dati da anni di lavoro da parte della comunità che si occupano della loro manutenzione. Le librerie wxWidgets supportano a loro volta OpenGL, lo standard di mercato per quanto riguarda la grafica tridimensionale. Grazie ad OpenGL è possibile produrre grafica 2D e 3D sfruttando al massimo l'accelerazione hardware messa a disposizione dalle moderne schede video. Qualora l'accelerazione hardware non fosse disponibile, OpenGL ovviamente sopperisce a questa mancanza via software. In questo modo è stato possibile supportare la grafica ad alte prestazioni senza perdere assolutamente nulla in fatto di compatibilità e portabilità.

BIBLIOGRAFIA

¹ Grid Generation (prentice Hall)

² BOINC project Berkeley