



**TRASFERIMENTO DI
IMMAGINI TELERILEVATE
FRA SISTEMI ETEROGENEI**

Rapporto Interno C88-29

Maggio 1988

**Roberto della Maggiore
Elia Perotto**

**Trasferimento di Immagini Telerilevate
fra Sistemi Eterogenei**

Copyright Maggio 1988

**Roberto della Maggiore
Elia Perotto**

**Consiglio Nazionale delle Ricerche
Istituto CNUCE - Pisa**

Rapporto interno C88-29

Indice

Prologo	1
Introduzione	3
L' ambiente operativo	5
Problematica: da cosa a cosa	5
Supporti hardware e software	5
Modi operativi	6
Interfacciamento drivers	7
L' implementazione	9
Algoritmi di trasparenza	9
Su video o su disco	9
Trucchi di velocizzazione	10
Appendice	13

Prologo

Nell'ambito dell'elaborazione delle immagini un'esigenza primaria è il poter disporre di strumenti che consentano il trasferimento veloce delle immagini dai supporti di memorizzazione a quelli di visualizzazione. Il problema è particolarmente aggravato dal fatto che i files contenenti le informazioni che descrivono le immagini sono in genere di grosse dimensioni e comportano perciò tempi piuttosto lunghi per il trasferimento; inoltre, a peggiorare la situazione vi è anche la necessità di effettuare trasferimenti "trasparenti" perché i dati delle immagini sono dati binari puri, cioè utilizzano tutte le possibili configurazioni dell'unità elementare di informazione. È necessario perciò ricorrere a particolari accorgimenti per evitare

che i caratteri normalmente usati per il controllo della trasmissione e delle funzioni delle apparecchiature vengano interpretati come tali, ma possano invece giungere all'apparecchiatura di uscita, in maniera appunto "trasparente" alle altre apparecchiature intermedie. Per realizzare la trasparenza è necessario aumentare la quantità dei dati trasmessi e perciò tale requisito aggrava la situazione; l'aumento è dipendente dalle caratteristiche hardware delle apparecchiature e può comportare, nel peggiore dei casi, il raddoppio totale dei dati.

Nel seguito viene descritto come è stato affrontato e risolto il suddetto problema in ambiente VM per la rappresentazione di immagini su display grafico.

Introduzione

Il presente lavoro illustra i problemi incontrati e le soluzioni adottate nel collegamento tramite Personal Computer di un display grafico al sistema VM del CNUCE per la rappresentazione di immagini grafico-pittoriche.

Il supporto di collegamento utilizzato è la linea coassiale tipo 3270, utilizzata da macchina virtuale come device dedicato. Il Personal Computer consente inoltre la memorizzazione locale delle immagini, su disco o su diskette, e la successiva rappresentazione su display, con eventuale elaborazione locale. Un approccio diverso è stato seguito in un precedente lavoro

nell'ambito del quale il display è stato collegato tramite un'interfaccia seriale di tipo BSC, utilizzando un dispositivo hardware e software sviluppato appositamente nell'ambito del CNUCE e collegato al sottosistema RSCS. Questo modo di gestione del display presenta il vantaggio di essere utilizzabile anche a distanza mediante linea telefonica ma fornisce prestazioni inferiori in termini di tempo di trasmissione, rispetto a quello qui descritto, e soprattutto non consente la memorizzazione delle immagini su un supporto locale in grado di controllarle autonomamente.

L' ambiente operativo

Problematica: da cosa a cosa

L' esigenza primaria che ha mosso alla realizzazione del presente lavoro e' la necessita' di visualizzare immagini su un display grafico. Sul sistema VM del CNUCE e' da tempo disponibile un supporto di elaborazione, il CIPS, che consente la manipolazione di immagini registrate su memoria di massa in modo agevole e veloce, sfruttando le capacita' elaborative del mainframe. Il risultato di tali elaborazioni e' comunque un file di dati registrato su disco. Per la visualizzazione e' disponibile un supporto per il collegamento di un sistema grafico VDS 701 via linea seriale BSC. Questo costituisce una soluzione valida, abbastanza veloce (4 minuti circa per il trasferimento di un' immagine di 256k su linea a 9600 bps), ed ottima nel caso che la stazione grafica non si trovi nelle immediate vicinanze del mainframe, consentendo il collegamento TP via linea telefonica; pero' non e' in grado di fornire quelle prestazioni di manipolazione locale delle immagini che sono oggi consentite da stazioni costituite da un personal computer equipaggiato con scheda grafica e relativo monitor. Nella fattispecie e' stato percio' utilizzato un Personal Computer IBM compatibile, con sistema operativo MS-DOS, il quale costituisce una stazione grafica corredata con numerose funzioni locali per la manipolazione delle immagini.

Il supporto di trasmissione e' stato realizzato con doppia possibilita':

- ricezione dell' immagine direttamente su monitor se e' richiesta la visualizzazione immediata, in questo caso si ottiene il massimo della velocita';
- ricezione su disco o dischetto sia per creare un archivio locale di immagini

elaborate su mainframe, sia per avere la possibilita' di manipolare l' immagine sfruttando le capacita' della stazione grafica; in questo caso la velocita' di trasferimento e' condizionata da quella propria del supporto di memorizzazione locale.

Supporti hardware e software

Tre sono le componenti hardware utilizzate:

1. l' IBM 3081 gestito da sistema operativo VM e dotato di interfaccia coassiale 3270 di cui riportiamo i dati tecnici:
 - velocita': 1.25 (send) to 2.35 (receive) Mb/s
 - linea: cavo coassiale RG-62U da 93 ohm di impedenza
 - distanza: 1.5 Km
 - tasso di errore: molto basso (1.E-10)
2. il Personal Computer IBM o compatibile equipaggiato con interfaccia coassiale 3270 e con scheda grafica per la gestione del display; si sono utilizzate tre interfacce coassiali diverse (DCA-IRMA, CXI-PCOX e AttachMate) paragonando i diversi risultati; di esse si riportano i dati tecnici principali.
 - DCA-IRMA:
 - Processor: 8X305 Special Purpose Microprocessor + 8 Kb RAM
 - Trasferimento: Dual Port Register Array (220-227 H)

- Buffer: 3440 bytes (non mappati sul bus 8088)
- Emulazione: 3278 mod. 2, 3 e 4, 3279 mod. 2A e 3A
- CXI-PCOX:
 - Processor: Proprietary gate array
 - Trasferimento: DMA 1, DMA 3 oppure in emulazione IRMA
 - Buffer: sufficiente per 2 sessioni contemporanee
 - Emulazione: 3278 mod. 2, 3, 4 e 5, 3279 mod. 2A e 3A
- AttachMate 3-N-I:
 - Processor: dato non disponibile
 - Trasferimento: non necessario se non in emulazione IRMA
 - Buffer: mappato in memoria PC
 - Emulazione: 3278 mod. 2, 3, 4 e 5, 3279 mod. 2A e 3A, 3290 e 3287

La scheda grafica utilizzata e' la Revolution 32 prodotta dalla Number Nine Computer Corporation che nella configurazione utilizzata consente di stampare immagini di 512x512 pixels;

3. il display grafico ad alta risoluzione: sono stati utilizzati monitors di varie case costruttrici (Barco, Mitsubishi, Philips); la scelta fra l'uno o l'altro e' un fattore puramente economico in quanto la differenza consiste solo nella resa grafica dell'immagine, almeno per cio' che riguarda i fini di questo lavoro.

Modi operativi

In ambiente VM e' possibile utilizzare una linea coassiale 3270 in due maniere distinte:

- sotto il diretto controllo del Sistema Operativo centrale, come console di macchina virtuale,
- sotto il controllo di un programma applicativo, come device dedicato ad una macchina virtuale.

Per la realizzazione del lavoro e' stato scelto quest'ultimo modo di operare. Nel primo caso il VM fornisce il supporto per l'uso della linea come apparecchiatura di collegamento di un terminale 3270 (nei vari modelli); e' questo un sistema abbastanza sofisticato che comporta la formattazione dello schermo ed il rispetto di alcune regole dovute all'uso specifico della consolle. La linea coassiale in questo caso viene gestita direttamente dal sistema e l'accesso ad essa da parte di un programma applicativo e' reso possibile mediante l'uso di apposite macro, a meno di non volersi sobbarcare un carico di gestione abbastanza pesante. Un vantaggio derivante da questo modo di utilizzare la linea 3270 e', nel nostro caso, che risulta sufficiente una sola linea, mentre ne servono invece due nell'altro caso, con in piu' un terminale per uso di console, oltre al personal della stazione grafica. Cio' nonostante si e' preferito optare per il secondo modo di utilizzazione in quanto esso consente una maggiore flessibilita' e semplicita' di uso e di programmazione; si ha inoltre un controllo piu' diretto sullo stato della linea, anche se e' necessario scendere ad un maggiore dettaglio di programmazione, gestendo direttamente le operazioni di I/O sulla linea.

Interfacciamento drivers

L'obiettivo principale da raggiungere in questo lavoro è la minimizzazione dei tempi di trasferimento delle immagini da host a workstation. Ciò si raggiunge non soltanto inviando in linea la minor quantità di dati possibile, ma anche riducendo il turnaround-time fra sistema e terminale e riducendo al minimo il numero dei turnaround. Così si è operato verso la prima direzione realizzando un'interfaccia molto specializzata ma efficiente, e verso la seconda utilizzando la più ampia dimensione possibile del buffer di scambio.

Il protocollo di interfacciamento del driver su host con quello su personal è molto semplice e rigido e non consentirebbe agevolmente un uso del sistema di comunicazione diverso da quello per cui è stato realizzato. Il protocollo consiste in pratica nell'uso di un carattere di parità associato ad ogni blocco trasmesso e di un carattere di controllo per indicare l'ultimo blocco della trasmissione. Ogni blocco viene confezionato dal trasmettitore e corredato con l'indicazione della lunghezza del blocco stesso. La sincronizzazione fra trasmettitore e ricevitore è assicurata da un carattere di acknowledgement inviato dalla workstation al completamento della

ricezione del buffer. Dato il tipo di collegamento non sono previste situazioni di hang-up per le quali sia necessaria una ritrasmissione o comunque una azione di recovery. Questo protocollo di sincronizzazione minimo è stato poi leggermente modificato nell'intento di ottenere una maggiore velocità di trasferimento realizzando un certo grado di quasi-parallelismo fra l'acquisizione del dato dalla linea ed il suo trasferimento sul supporto locale di utilizzazione (vedere a questo proposito il capitolo relativo alla velocizzazione).

Per quanto riguarda le dimensioni del buffer si è utilizzata un'interfaccia 3270 mod. 4 con buffer di 3.5 kbytes; a questo proposito si è osservato sperimentalmente che una delle componenti più penalizzanti in questo scambio di dati è il tempo di risposta del VM, non in termini di tempo di elaborazione per la preparazione dei dati (che risulta trascurabile), ma in termini di dispatching. Infatti un miglioramento notevole dei tempi di attesa del VM si è avuto passando dal buffer di dimensioni minori (3270 mod. 2, buffer di 2 kbytes) a quello di dimensioni maggiori, minimizzando così il numero di blocchi trasmessi (vedere i dati statistici in appendice).

L' implementazione

Algoritmi di trasparenza

Anche per la realizzazione della trasparenza l'obiettivo principale da raggiungere e' l'efficienza della trasmissione in termini di tempo, soprattutto di tempo di elaborazione sul personal che e' il componente piu' critico del sistema. L'uso dell'interfaccia 3270 si e' rivelata particolarmente penalizzata nel caso di trasmissione trasparente, rispetto a tecniche di trasmissione diverse quali il BSC o l' SDLC in quanto il numero di caratteri che necessitano di un trattamento particolare per non perdere di significativita' e' nettamente superiore, circa un quarto dell'intero set, anche se poi solo una piccola parte di essi viene effettivamente utilizzata per scopi di controllo della trasmissione. Questo rappresenta una limitazione imposta dalle caratteristiche di definizione dell'interfaccia 3270 che non e' possibile scavalcare a livello applicativo. Si e' pensato in un primo tempo di effettuare un rimappamento dei caratteri in modo da trovarsi il meno possibile nella necessita' di gestire caratteri trasparenti. Constatato tuttavia che una buona parte del sottoinsieme dei caratteri da trattare risulta statisticamente fra i meno utilizzati nella trasmissione delle immagini si e' proceduto direttamente a renderli trasparenti senza operare il rimappamento dell'intero set, operazione questa che avrebbe un certo costo in termini di tempo di esecuzione. Dato che la control unit 3274 traduce i byte nel codice Display 3274 abbiamo dovuto prima tradurre i dati con la tabella inversa.

Per quanto riguarda in dettaglio il modo di effettuare la trasmissione dei caratteri "non buoni" si e' dovuto agire in un modo un po' piu' complicato di quello classico che prevede l'uso di un opportuno carattere di controllo da premettere al carattere incriminato per segnalare che esso e' "dato" e

non "controllo". Cio' perche' si e' operato non a livello di control unit ma a livello applicativo ed inoltre nel caso del 3270 la maggior parte dei caratteri "non buoni" non vengono utilizzati come controllo, ma solamente tradotti dalla control unit in un unico carattere non significativo perche' non esistono nel codice Display 3270. Pertanto per i soli caratteri che fanno parte di questo sottoinsieme si e' dovuto provvedere ad un'operazione di raddoppio piu' rimappamento. Altri algoritmi per la realizzazione della trasparenza, come ad esempio quello che prevede due bytes di informazione su tre bytes effettivamente trasmessi e che e' molto usato in ambiente 3270, sono stati scartati perche' da un calcolo effettuato a priori presentano l'inconveniente di effettuare la trasmissione di un numero di caratteri mediamente superiore. Comportano pertanto un maggiore onere di elaborazione risultando quindi globalmente meno efficienti di quello implementato.

Su video o su disco

Come gia' accennato il supporto di ricezione dell'immagine su personal e' stato realizzato in due versioni distinte: una consente l'uso del display grafico come monitor collegato direttamente al VM; in questo caso il personal e' gestito da un programma che si comporta in pratica come uno stand-alone e fa svolgere al personal stesso la funzione di interfaccia passiva per la gestione del monitor. Con questo tipo di gestione si raggiunge la massima velocita' di trasferimento possibile che e' di poco inferiore a quella propria del VM per la gestione dei terminali 3270.

L'altro programma invece non effettua direttamente il trasferimento al monitor

dell' immagine ricevuta, ma la memorizza su supporto locale (diskette o hard disk), ed e' pertanto vincolato alla velocita' di quest' ultimo. Questa versione consente la creazione di un archivio locale di immagini precedentemente elaborate su mainframe e necessita di altro software per la visualizzazione sul display grafico di tali immagini; questo supporto costituisce pertanto uno strumento integrativo delle workstation grafiche per elaborazione di immagini sviluppate dal CNUCE su Personal Computer.

Si osserva che una volta registrata l' immagine su memoria di massa locale non si e' piu' legati all' hardware descritto precedentemente per la rappresentazione, ma e' possibile utilizzare schede grafiche diverse con relativo software.

Trucchi di velocizzazione

Al fine di rendere piu' veloce la trasmissione si e' agito su entrambe le componenti interessate: il mainframe e il micro.

Per quanto riguarda il primo e' bene fare un' osservazione preliminare riguardo al sistema operativo: essendo il VM un sistema time-sharing le possibilita' di intervento sono limitate all' applicazione che viene fatta funzionare sulla macchina virtuale; non e' possibile invece intervenire sui tempi di dispatching della macchina stessa se non in maniera limitata e comunque solo da parte del gestore del sistema e non agendo sul programma applicativo.

E' da considerare inoltre che i tempi di esecuzione del mainframe (nel caso l' IBM 3081) sono di alcuni ordini di grandezza inferiori a quelli del personal e rendono percio' inutile una grande raffinatezza di programmazione che viene poi vanificata dai gia' citati aspetti software del sistema operativo. Pertanto l' intervento operato sul programma lato VM e' abbastanza limitato e si rivolge in due direzioni distinte: da una parte, guardando all' efficienza del programma residente su mainframe e' stato implementato un algoritmo di bootstrapping che consente di avere gia' disponibile un blocco di dati da trasmettere non appena il partner ne fa richiesta, dall' altra tenendo in considerazione le prestazioni inferiori del personal, si e' provveduto a svolgere sul mainframe le operazioni

piu' indicate per farlo lavorare il meno possibile.

La considerazione su cui ci si e' basati per realizzare cio' e' che in una normale trasmissione 3270 il terminale riceve i caratteri codificati secondo il codice proprio del 3270 stesso (3270 scan code) e ad essi viene assegnato l' esatto significato da una tabella di decodifica opportuna residente sul terminale o, nel caso di emulatore, sul personal. In pratica percio' l' intervento si e' concretizzato nel far confezionare il buffer di trasmissione con caratteri filtrati attraverso una opportuna tabella inversa di trascodifica, in modo che il ricevente ottenga direttamente un buffer di dati gia' pronti all' uso e sia cosi' sgravato dalla necessita' di effettuare traduzioni.

Per quanto riguarda invece la soluzione adottata sul personal per ottenere una maggiore velocita' nella ricezione delle immagini, l' accorgimento principale e' scaturito empiricamente col procedere delle prove di trasmissione.

Innanzitutto si e' verificato che la tecnica di ricezione dei caratteri migliore per la nostra applicazione e' quella implementata dalla scheda IRMA e che consiste nella lettura di un carattere per volta dal buffer di ricezione, anche se a prima vista possono sembrare vincenti tecniche diverse quali ad esempio il mappamento in memoria del buffer di ricezione, effettuabile con la scheda Attachmate.

Si e' poi osservato misurando da programma i tempi di elaborazione, che una buona percentuale del tempo viene impiegata nell' attesa per la ricezione dell' intero buffer (i due partner si sincronizzano mediante un carattere indicante la parita' del blocco di dati). Questo dato e' piuttosto sorprendente, considerata la natura del canale di collegamento fra i due sistemi che e' la linea coassiale, la quale ha una velocita' nominale di 10 Mbit/sec. Tuttavia i risultati ottenuti in numerose prove di trasmissione hanno mostrato che la velocita' effettiva e' di gran lunga inferiore.

Si e' allora provveduto ad inserire i caratteri di sincronizzazione di blocco non solo alla fine dell' intero buffer, ma anche in posizioni intermedie, frantumando cosi' il buffer in un numero opportuno di record (empiricamente stabilito in 10). In tal modo si e' potuto ottenere un certo grado di

parallelismo fra le operazioni di riempimento del buffer da parte dal mainframe e quelle di svuotamento del medesimo da parte del personal: infatti quest'ultimo è in grado di iniziare le operazioni di lettura

e trasferimento non appena gli viene reso disponibile un record, e queste operazioni possono aver luogo mentre il mainframe sta continuando a trasmettere il resto del blocco di dati.

Appendice

File Kbytes	wait VM	wait PC	Disk	tempo sec.	vel. Kbit/s	IBM Prog.	descrizione
131.	28.9	8.3	24.7	61.9	16.9	SX6	IRMA on IBM PC
131.	28.3	8.4	24.9	61.6	17.	SX6	
131.	29.7	8.2	24.	61.9	16.9	SX6	Olivetti M24
131.	29.	8.7	24.3	62.	16.9	SX6	
131.	28.3	8.2	24.8	61.3	17.1	SX6	Olivetti M24
131.	30.1	7.9	26.4	64.4	16.3	SX6	Olivetti M24
129.	1.9	10.3	25.	37.2	27.8	Screen8	M24 with PCOX card (IRMA emul.)
131.	5.3	31.7	25.6	62.6	16.7	SX6	M24, VM wait = da linea a schermo
131.	22.4	14.4	25.4	61.8	17.	SX6	M24
131.	1.3	35.8	25.9	63.	16.6	SX6	M24, VM wait = da linea a schermo
131.	51.6	8.7	24.7	85.	12.3	SX6	M24, VM wait alto (sistema carico)
129.	3.9	10.4	25.7	40.	25.9	Screen8	M24 with PCOX card (IRMA emul.)
131.	33.9	9.8	25.2	68.9	15.2	SX6	M24
129.	3.8	10.1	25.6	39.5	26.2	Screen8	M24 with PCOX card (IRMA emul.)
129.	3.3	13.7	6.	23.	45.	Screen8	M24 - Graf F28
131.	31.4	10.5	5.2	47.1	22.3	SX6	M24 - Graf F28 (fast hard disk)
131.	32.7	11.	4.9	48.6	21.6	SX6	M24 - Prog.ITX6X on VM (optim.)
131.	41.7	1.8	0.	43.5	24.1	SX6	M24 - No disk write
131.	41.6	0.	0.	41.6	25.2	SX6	M24 - No PC proc., no disk write
129.	9.3	14.9	4.8	29.	35.7	Screen8	M24 - IRMA DSI card
129.	7.2	13.6	5.	25.8	40.1	Screen8	M24 - PCOX emulating IRMA
129.	6.7	13.9	5.1	25.7	40.3	Screen8	M24 - PCOX emulating IRMA
131.	31.6	11.	4.3	46.9	22.4	SX6	M24
262.	23.5	16.9	-.	40.4	51.9	GTM	IBM AT/AV to Screen
262.	19.	17.	6.5	42.5	49.4	IT	IBM AT/AV to Hard Disk
262.	18.6	17.	4.5	40.1	52.3	IT	PC-Bit 286, DCA-IRMA, Turbo
262.	19.4	17.	4.	40.4	51.9	IT	PC-Bit 286, DCA-IRMA, Turbo
262.	15.1	21.4	4.7	41.2	50.9	IT	PC-Bit 286, DCA-IRMA
262.	14.4	21.6	4.4	40.4	51.9	IT	PC-Bit 286, DCA-IRMA
262.	15.3	21.3	4.5	41.2	50.9	IT	PC-Bit 286, DCA-IRMA

