Cluster di PC per calcolo teorico

Francesco Zuliani

Dipartimento di Fisica
INFN di Cagliari

in collaborazione con E. Marinari, A. Chessa, A. Fara

COFIN (Cagliari, Pisa, Roma1, Roma2)

francesco.zuliani@ca.infn.it

Ottobre 1999 Taranto, Martina Franca

Teoria dei Campi

Grazie alla connessione tra Meccanica Statistica e Teoria dei Campi si possono trattare con metodi numerici di tipo Monte Carlo domande fondamentali in diverse discipline della Fisica Teorica

- Spettro di massa e ampiezze di decadimento nelle teorie di *gauge* su reticolo.

- Rottura Spontanea della Simmetria di Replica nei sistemi disordinati

- Gravità quantistica

Relazioni tra problema fisico e tipo di hardware

- Il tipo di problemi numerici che uno si trova ad affrontare sono leggermente diversi a seconda dello specifico sistema fisico.

- In tutti i problemi la necessità comune è di simulare delle equazioni, in genere locali, su un reticolo d-dimensionale.

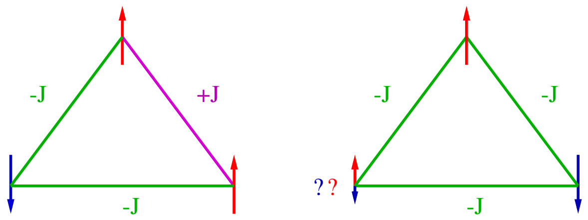
- La grandezza del reticolo e il tipo di variabili che si devono aggiornare giocano un ruolo chiave nella scelta delle risorse ottimali.

Teorie di *gauge*

- In ogni punto e *link* del reticolo le variabili fondamentali sono i campi fermionici e bosonici. Essenzialmente matrici di variabili complesse.
- L'aggiornamento secondo le equazioni di evoluzione prevede il calcolo di somme e prodotti di matrici complesse. L'aggiornamento è decisamente CPU *intensive*.
- Comparativamente ai sistemi disordinati si riescono a termalizzare reticoli relativamente grandi. La quantità di memoria richiesta per sito e complessiva è molto grande.
- La quantità di CPU e memoria richiesta vanno sempre oltre le capacità di una tipica workstation.

Teorie di campo dei sistemi disordinati: il caso dei vetri di spin

- Il disordine nella hamiltoniana dei vetri di spin induce frustrazione, che è sinonimo di degenerazione e metastabilità.



- Le normali simulazioni all'equilibrio hanno dei tempi di rilassamento esponenziali nel volume. In genere i reticoli da simulare sono relativamente piccoli.
- Le variabili di sito sono semplici variabili di spin $\{+1; -1\}$. Le variabili di *link* possono essere sia intere che reali.
- La dinamica di evoluzione per spin è computazionalmente leggera ad ogni passo.
- I risultati devono essere mediati sul disordine.

I Cluster di PC: cluster vs. farm

I Cluster

La soluzione consiste nella suddivisione del reticolo in sottoreticoli da associare a nodi di calcolo differenti. Questa soluzione è abbastanza soddisfacente tenendo conto che per un'interazione locale ci si aspetta che la richiesta di CPU $\sim L^d$ mentre la necessità di comunicare le interazioni sui bordi $\sim L^{d-1}$. La comunicazione gioca tuttavia un ruolo importante.

Le Farm

Ai vari nodi vengono assegnate replicazioni del sistema originario. Per aumentare la statistica o per simulare le diverse realizzazioni del disordine su cui fare le medie.

Kalix2: l'hardware

1 Server

- Mainboard PII Triton ATX AGP ASUS P2B-F
- 1 CPU Pentium II 450 Mhz (15%)
- Memoria: 256Mb (20%)
- Scheda Grafica: Matrox G200 SD - 8Mb - AGP
- Hard Disk 8,4Gb Eide Ultra DMA 5400 RPM (6%)
- Masterizzatore: Ide RW HP 8100 (2x 4x 24x) (10%)
- Tape: Ditto Max Pro Interno
- 2 Schede Ethernet PCI 10/100 DIGITAL DE500
- Monitor 17″ SONY 200 GST (18%)
- Floppy, ventoline, case, assemblaggio, scheda audio.

Costo totale 4.756.000.

Kalix2: l'hardware
. . . continued

16 Client

- Mainboard Dual PII (SuperMicro P6DBE, 5PCI, 2ISA, 100MHz) (12%)
- 2 CPU Pentium II 450 Mhz (37%)

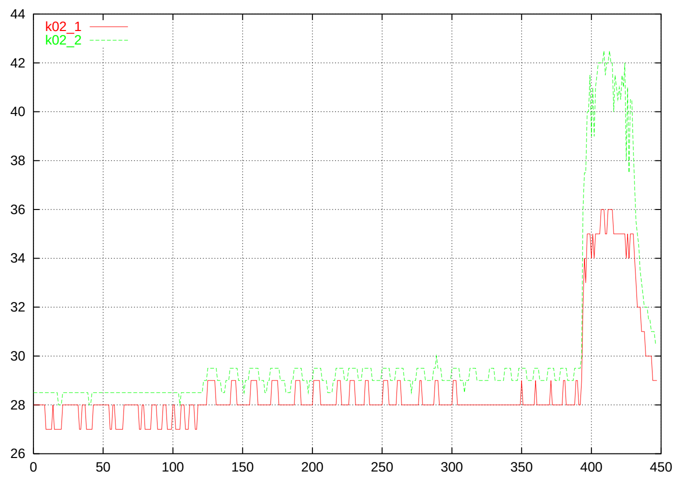
  Migliore gestione latenze o CPU intensive
- Memoria: 256Mb (25%)
- Scheda Grafica: Intel 740
- Hard Disk 8,4Gb Eide Ultra DMA 5400 RPM (8%)
- 3 Schede Ethernet PCI 10/100 DIGITAL DE500 (15%)
- Floppy, ventoline, case, assemblaggio

Costo totale 3.734.000 x 16 = 59.744.000.

HUB 3Com 3300 (24 porte) 100Mb/s (4.350.000)

Kalix2: Caveat sull'hardware

- SMP scaldano MOLTO ($30 - 50°C$)!
  Ambiente refrigerato



CPU a riposo, al tempo 380 iniziano a girare programmi

- Memorie non scadenti (...altrimenti signal 11).

  La verifica più stringente si ottiene compilando tanti alberi del kernel contemporaneamente, per saturare la memoria. (Usare anche l'opzione '-j 5' nel Makefile della root)

  I programmi tipo 'memtest' non sono da soli sufficienti.

  (`http://reality.sgi.com/cbrady_denver/memtest86/`)

- Attualmente in sala calcolo sotto gruppo di continuità

  (Potenza stimata da quella dell'alimentatore $\approx$ 200W)

Kalix2: il software

- Linux, RedHat 6.0 (Kernel 2.2.9 (nfs) ricompilato *ad hoc*)
- Gestione centralizzata (Le root directory dei client sono sul server (Netboot))
- Software centralizzato
- Output del comando 'df' sul client

```
SERVER-IP:/home/clients/k06
                   5576525   2515945   2771814  48% /
SERVER-IP:/usr     1981000   1127918    750670  60% /usr
SERVER-IP:/sbin    202182    162010      29732  84% /sbin
SERVER-IP:/home    5576525   2515945   2771814  48% /home
SERVER-IP:/lib     202182    162010      29732  84% /lib
SERVER-IP:/bin     202182    162010      29732  84% /bin
/dev/hda7          7639582   4423549   2820033  61% /local_disk
```

- Output del comando 'df' sul server

```
/dev/hda5             202182    162011     29731  84% /
/dev/hda8            5576525   2515972   2771787  48% /home
/dev/hda7            1981000   1127918    750670  60% /usr
k01:/local_disk      7639582    400819   6842763   6% /k01
...
k16:/local_disk      7639582    477944   6765638   7% /k16
```

Kalix2: il software (parte II)

- lm_sensors (Temperature Ventole)
- MPICH (funziona con rsh quindi ricordarsi /etc/hosts.equiv)

  (anche per gestire la doppia CPU)
- PVM
- Threads per gestire la doppia CPU
- Schede Digital perché GAMMA (libreria a bassa latenza)

  Da poco hanno implementato MPI (versione MPI-CH) su GAMMA (ancora non l'abbiamo provato ma . . . , si basa sull'implementazione degli "Active Messages")

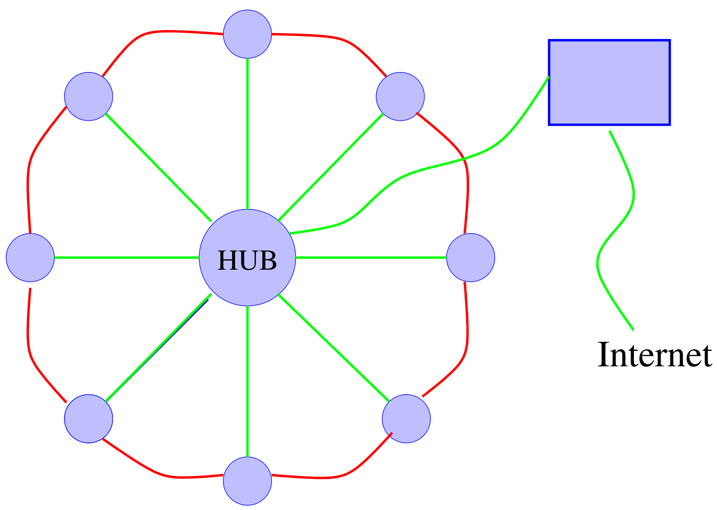
  `http://www.disi.unige.it/project/gamma`

Kalix2: Topologia della rete

Incidendo sul per il 10% del prezzo complessivo di un nodo abbiamo ritenuto che fosse un'esperimento degno di essere portato avanti.

In fisica le interazioni sono a primi vicini.

MPI permette modalità di comunicazione molto complicate.



Kalix2: Il NetBoot

Lato Client

- compilazione del Kernel con abilitato:

```
CONFIG_IP_PNP=y         (kernel level autoconfiguration)
CONFIG_IP_PNP_BOOTP=y   (modelita‘ di autoconfigurazione)
CONFIG_DE4X5=y          (la scheda di rete non puo‘ essere un modulo)
CONFIG_NFS_FS=y         (NFS non puo‘ esser un modulo)
CONFIG_ROOT_NFS=y       (NFS-root va abilitato esplicitamente)
```

- **`netboot-0.8.1`**

  contiene un wrapper dell'immagine del kernel

  ```
  mknbi-linux -d rom -i rom -k zImage -o bootImage229-client-1
  ```

  che in realtà risiede nel server e viene caricata tramite (tftp)

  Con il comando **`make bootrom`**, usando i packet driver sotto DOS, si creano delle immagini (image.flo image.hex) da usare rispettivamente su un floppy (anche da Hard-disk) da flashare sulla EEPROM

- Usando il boot da EEPROM bisogna abilitarlo nel BIOS.

Kalix2: Il NetBoot (Parte II)

Lato Server

- **bootpd (DHCPD)**
- **/etc/inetd.conf**

```
tftp    dgram   udp     wait    root    /usr/sbin/tcpd
                                               in.tftpd /home/clients
bootps  dgram   udp     wait    root    /usr/sbin/tcpd  bootpd
```

- **/etc/exports**

```
/usr                      *.kalix2.unica.it(ro,no_root_squash)
/sbin                     *.kalix2.unica.it(ro,no_root_squash)
/bin                      *.kalix2.unica.it(ro,no_root_squash)
/lib                      *.kalix2.unica.it(rw,no_root_squash)
/home                     *.kalix2.unica.it(rw,no_root_squash)
/clients/COMMONfiles      *.kalix2.unica.it(ro,no_root_squash)
```

- **/etc/bootptab**

```
 client1:hd=/home/clients:vm=auto:ip=k01:\
        :ht=ethernet:ha=0000f8099969:\
        :bf=bootImage229-client-1:rp=/home/clients/k01
...
client16: ....
```

- **/etc/rc.d/init.d/nfs**

Modificare RPCNFSDCOUNT=18 in modo che dal lato server ci siano 18 (e non 8, il default) demoni in ascolto per le richieste di NFS

- Se si usano i tcp_wrapper (tcpd) bisogna abilitare in `/etc/hosts.allow`

  la broadcast per il bootpd cioè:

  ```
  bootps: 0.0.0.0
  ```

  Per NFS bisogna abilitare portmap sul Numero IP dei client

  ```
  portmap: 192.168.100.1
  portmap: 192.168.100.2
  ...
  portmap: 192.168.100.16

  rpc.mountd: k01, k02, .... , k16
  rpc.nfsd: k01, k02, .... , k16
  ```

Struttura delle directory client, sul server

La lista di eseguibili minimale per le root-dir dei client sul server è

```
-rwxr-xr-x   1 root     root   k01/sbin/ifup
-rwxr-xr-x   1 root     root   k01/sbin/init
-rwxr-xr-x   1 root     root   k01/sbin/initlog
-rwxr-xr-x   1 root     root   k01/sbin/loglevel
lrwxrwxrwx   1 root     root   k01/sbin/swapoff -> swapon
-rwxr-xr-x   1 root     root   k01/sbin/swapon
-rwxr-xr-x   1 root     root   k01/sbin/route
-rwxr-xr-x   1 root     root   k01/sbin/portmap
-rwxr-xr-x   1 root     root   k01/sbin/ifconfig
-rwxr-xr-x   1 root     root   k01/sbin/halt
lrwxrwxrwx   1 root     root   k01/sbin/reboot -> halt
drwxr-xr-x   2 root     root   k01/lib
-rwxr-xr-x   1 root     root   k01/lib/ld-2.1.1.so
lrwxrwxrwx   1 root     root   k01/lib/ld-linux.so.2 -> ld-2.1.1.so
-rwxr-xr-x   1 root     root   k01/lib/libc-2.1.1.so
lrwxrwxrwx   1 root     root   k01/lib/libc.so.6 -> libc-2.1.1.so
-rwxr-xr-x   1 root     root   k01/lib/libutil-2.1.1.so
lrwxrwxrwx   1 root     root   k01/lib/libutil.so.1 -> libutil-2.1.1.so
lrwxrwxrwx   1 root     root   k01/lib/libtermcap.so.2 -> libtermcap.so.2.0.8
-rwxr-xr-x   1 root     root   k01/lib/libtermcap.so.2.0.8
-rwsr-xr-x   1 root     root   k01/bin/mount
-rwsr-xr-x   1 root     root   k01/bin/umount
-rwxr-xr-x   1 root     root   k01/bin/rm
-rwxr-xr-x   1 root     root   k01/bin/sh
-rwxr-xr-x   1 root     root   k01/bin/hostname
lrwxrwxrwx   1 root     root   k01/bin/domainname -> hostname
-rwxr-xr-x   1 root     root   k01/bin/loadkeys
-rwxr-xr-x   1 root     root   k01/bin/kill
```

`/dev` una copia per ogni client, ma forse andrebbero bene anche degli hard-link per non replicare le cose inutili. Inoltre si potrebbere togliere i device che non servono, ad esempio gli scsi se uno ha solo cose ide, etc etc

File del client k01 sul server in `/home/clients/k01` la directory `etc` ha alcuni file veri (quelli diversi in ogni macchina), altri sono link a file in `/k254/etc` che viene montata a boot (vedi `/etc/fstab` dopo)

```
/etc ad esempio
-rw-rw-r--   1 root     root          606 Jul  9 10:54 k01/etc/fstab
-rw-r--r--   1 root     root          724 Jul  8 15:29 k01/etc/hosts
-rw-r--r--   1 root     root         1756 May 13 15:07 k01/etc/inittab
drwxr-xr-x  10 root     root         1024 Jun  2 20:46 k01/etc/rc.d
```

client /etc/fstab

```
/dev/hda5               swap                    swap    defaults        0 0
/dev/hda6               swap                    swap    defaults        0 0
/dev/hda7               /local_disk             ext2    defaults,usrquota  1 1
192.168.100.254:/home/clients/k01       /               nfs     defaults   1 1
192.168.100.254:/bin                    /bin            nfs
192.168.100.254:/usr                    /usr            nfs
192.168.100.254:/sbin                   /sbin           nfs
192.168.100.254:/home                   /home           nfs
192.168.100.254:/lib                    /lib            nfs
none                    /proc                   proc    defaults        0 0
none                    /dev/pts                devpts  mode=0622       0 0
```

Il file /etc/rc.d/rc.sysinit va modificato per montare al più presto le directory del server. Quello che segue è una variante di quello standard di RedHat6.0

```
#!/bin/sh
#
# /etc/rc.d/rc.sysinit - run once at boot time
#
# Taken in part from Miquel van Smoorenburg's bcheckrc.
#

# Rerun ourselves through initlog
if [ -z "$IN_INITLOG" ]; then
 [ -f /sbin/initlog ] && exec /sbin/initlog -r /etc/rc.d/rc.sysinit
fi

# Set the path
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

# Read in config data.
if [ -f /etc/sysconfig/network ]; then
    . /etc/sysconfig/network
else
    NETWORKING=no
    HOSTNAME=localhost
fi

# Source functions
. /etc/rc.d/init.d/functions

# Fix console loglevel
/sbin/loglevel $LOGLEVEL

# Load keymap
KEYMAP=
if [ -f /etc/sysconfig/console/default.kmap ]; then
  KEYMAP=/etc/sysconfig/console/default.kmap
else
  . /etc/sysconfig/keyboard
  if [ -n "$KEYTABLE" -a -d "/usr/lib/kbd/keymaps" ]; then
     KEYMAP=$KEYTABLE
  fi
fi
if [ -n "$KEYMAP" ]; then
```

```
  # Since this takes in/output from stdin/out, we can't use initlog
  echo -n "Loading default keymap"
  loadkeys $KEYMAP < /dev/tty0 > /dev/tty0 2>/dev/null && \
     success "Loading default keymap" || failure "Loading default keymap"
  echo
fi

# Start up swapping.
action "Activating swap partitions" swapon -a

# Set the hostname.
action "Setting hostname ${HOSTNAME}" hostname ${HOSTNAME}

# Set the NIS domain name
if [ -n "$NISDOMAIN" ]; then
    action "Setting NIS domain name $NISDOMAIN" domainname $NISDOMAIN
else
    domainname ""
fi

if [ -f /fsckoptions ]; then
fsckoptions=`cat /fsckoptions`
    else
fsckoptions=
fi

if [ -f /forcefsck ]; then
fsckoptions="-f $fsckoptions"
fi

_RUN_QUOTACHECK=0
if [ ! -f /fastboot ]; then
        STRING="NOT Checking root filesystem, (NFS-root)"
echo $STRING
fi

# check for arguments

mount -t proc /proc /proc

# Remount the root filesystem read-write.
action "Remounting root filesystem in read-write mode" mount -n -o remount,rw /


if [ -n "$IN_INITLOG" ]; then
```

```
    IN_INITLOG=
fi

if [ ! -f /etc/HOSTNAME ]; then
    echo ${HOSTNAME} > /etc/HOSTNAME
fi

# Clear mtab
>/etc/mtab

# Enter root and /proc into mtab.
mount -f /
mount -f /proc

rm -f /lib/modules/preferred
rm -f /lib/modules/default

action "Bringing up interface lo" ifconfig lo 127.0.0.1 up

echo -n "Starting portmapper: "
portmap

echo
> /var/lock/subsys/portmap

action "Mounting NFS server usr filesystems" mount /usr
action "Mounting NFS server sbin filesystems" mount /sbin
action "Mounting NFS server home filesystems" mount /home
action "Mounting NFS server lib filesystems" mount /lib
action "Mounting NFS server bin filesystems" mount /bin


# Check local filesystems
if [ ! -f /fastboot ]; then
        STRING="Checking filesystems"
echo $STRING
#
# option on fsck means -A tutti quelli nonauto, -T no title  (poco verboso)
# -R noroot file system!!! -a automatic repair $fsckoption dovrebbe
# essere vuota
initlog -c "fsck  -T -R -A -a $fsckoptions"
rc=$?
        if [ "$rc" = "0" ]; then
success "$STRING"
echo
```

```
elif [ "$rc" = "1" ]; then
        passed "$STRING"
echo
fi

# A return of 2 or higher means there were serious problems.
if [ $rc -gt 1 ]; then
        failure "$STRING"
echo
echo
echo "*** An error occurred during the file system check."
echo "*** Dropping you to a shell; the system will reboot"
echo "*** when you leave the shell."

PS1="(Repair filesystem) \#"; export PS1
sulogin

echo "Unmounting file systems"
umount -a
mount -n -o remount,ro /
echo "Automatic reboot in progress."
reboot
elif [ "$rc" = "1" -a -x /sbin/quotacheck ]; then
# se fsck ha fatto qualche riparazione ricalcola la quota!!
action "Checking filesystem quotas" /sbin/quotacheck -v -R -a
fi
fi

# Mount all other filesystems (except for NFS and /proc, which is already
# mounted). Contrary to standard usage,
# filesystems are NOT unmounted in single user mode.

action "Mounting local filesystems" mount -a -t nonfs,smbfs,ncpfs,proc


if [ -x /sbin/quotaon ]; then
    action "Turning on user and group quotas for local filesystems" /sbin/quotaon -a
fi

# A questo punto tutti i file system sono montati e quindi
# /tmp e /var dovrebbero puntare a directory locali !!!
# Clean out /etc.
rm -f /etc/mtab~ /fastboot /fsckoptions /forcefsck
>/var/run/utmp
touch /var/log/wtmp
```

```
chgrp utmp /var/run/utmp /var/log/wtmp
chmod 0664 /var/run/utmp /var/log/wtmp

# Delete pam_console lock and refcount files
rm -f /var/lock/console.lock
rm -f /var/lock/console/*

# Delete UUCP lock files.
rm -f /var/lock/LCK*

# Delete stale subsystem files.
rm -f /var/lock/subsys/*

# Delete stale pid files
rm -f /var/run/*.pid

# Delete X locks
rm -f /tmp/.X*-lock

# Delete Postgres sockets
rm -f /tmp/.s.PGSQL.*

# Set the system clock.
ARC=0
UTC=0
if [ -f /etc/sysconfig/clock ]; then
    . /etc/sysconfig/clock

    # convert old style clock config to new values
    if [ "${CLOCKMODE}" = "GMT" ]; then
    UTC=true
    elif [ "${CLOCKMODE}" = "ARC" ]; then
    ARC=true
    fi
fi

CLOCKDEF=""
if [ -x /sbin/hwclock ]; then
    CLOCKFLAGS="--hctosys"
    CLOCK=/sbin/hwclock
    $CLOCK --adjust
else
    CLOCKFLAGS="-a"
    CLOCK=/sbin/clock
fi
```

```
case "$UTC" in
  yes|true)
    CLOCKFLAGS="$CLOCKFLAGS -u";
    CLOCKDEF="$CLOCKDEF (utc)";
  ;;
esac

if [ "$CLOCK" = "/sbin/clock" ]; then
 case "$ARC" in
   yes|true)
     CLOCKFLAGS="$CLOCKFLAGS -A";
     CLOCKDEF="$CLOCKDEF (arc)";
   ;;
 esac
fi

$CLOCK $CLOCKFLAGS
action "Setting clock $CLOCKDEF: `date`" date

# Right, now turn on swap in case we swap to files.
swapon -a >/dev/null 2>&1
action "Enabling swap space" /bin/true


# Set preferred X display manager link
preferred=
if [ -f /etc/sysconfig/desktop ]; then
if [ -n "`grep GNOME /etc/sysconfig/desktop`" ]; then
preferred=gdm
elif [ -n "`grep KDE /etc/sysconfig/desktop`" ]; then
preferred=kdm
elif [ -n "`grep AnotherLevel /etc/sysconfig/desktop`" ]; then
preferred=xdm
fi
fi
if [ -n "$preferred" ] && which $preferred >/dev/null 2>&1; then
ln -snf ../..`which $preferred` /etc/X11/prefdm
else
        if [ ! -L /etc/X11/prefdm ]; then
if which gdm >/dev/null 2>&1; then
ln -snf ../..`which gdm` /etc/X11/prefdm
elif which kdm >/dev/null 2>&1; then
ln -snf ../..`which kdm` /etc/X11/prefdm
elif which xdm >/dev/null 2>&1; then
```

```
ln -snf ../..`which xdm` /etc/X11/prefdm
fi
fi
fi

# Now that we have all of our basic modules loaded and the kernel going,
# let's dump the syslog ring somewhere so we can find it later
dmesg > /var/log/dmesg
```

Questo testo è accessibile via WWW in

http://chimera.roma1.infn.it:80/TALKS/
1999.10.ZULIANI.MARTINAFRANCA/