

## Parallel execution guide

In order to use the parallel version of the Ion Implant Simulator (IIS) you need to have some previously installed software in all the computers you want to run concurrently the code. This text is intended for use with linux/unix systems mainly, but you can guess what to do to use windows systems.

1.- **RSH/RSHD**. You need a remote shell server daemon and a client. This software implements 3 network services: exec, login and shell. You must install the rsh/rshd software and start with inetd these 3 network services.

In order to work you need to permit the access without passwords adding a **.rhosts** file into your **\$HOME** directory at each host computer. The format of this file is:

```
Host1 user1
Host2 user2
...     ...
```

were Host1, Host2, etc. are the computers you will use, and user1, user2, etc. are the login names of the user accounts you want to use.

To test remote shell log in [user1@Host1](#) and run:

```
rsh Host1 -l user1 ls
```

the result will be the listing of the \$HOME directory of user1 at Host1 computer.

2.- **PVM**. Install Parallel Virtual Machine software. It is interesting to install the runtime environment and the developer suite. Also it is interesting to install **xpvm** utility (it needs perl) to debug the code.

In order to use this you could add a **.pvmrc** file into each **\$HOME** directory of each computer with a content like:

```
version
conf
ps
```

these commands will show you each time you start pvm the version, the hosts included in the virtual machine and the status of processes running in this virtual system.

To start the pvm you will need a **hostfile** to know what are the hosts that conform the virtual machine and the users and directories you will use. An example of hostfile will be:

```
ep=$HOME/Test
mortadelo    sp=1000 ep=~jesman/Test    lo=jesman    wd=~jesman/Test
filemon      sp=1000 ep=~antonio/Test2    lo=antonio   wd=~antonio/Test2
```

where **sp** indicates the relative speed of the computer, **ep** indicates the executables path, **wd** is the working directory and **lo** is the login on that computer.

## Example Test Suite

Uncompress the package and change the directory:

```
>tar xzvf TestPVM.tgz
>cd TestPVM
```

Then you must edit the **load** file in order to include the hosts you want to use and the number of processes to run at each host: In example:

```
mortadelo      1
filemon        1
bluebird-linux 1
```

After that you must configure the environment variables with:

```
>./Configure
```

this will create a file named **mortadelo.hostname** (if the local host is mortadelo). At this point you need to add the hosts you want to use with:

```
>AddHost      filemon      jesman      DirTest
>AddHost      bluebird-linux jesman      OtherDirTest
```

where filemon and bluebird-linux are the host names and jesman is the login account (remember the permissions with the rsh command) and finally DirTest and OtherDirTest are two possible directory names to create and copy all the needed files.

The time to configure **pvm** has gone. First, shutdown the pvm starting the program and halting:

```
>pvm
pvm> halt
```

And now start pvm with the new and correct configuration and exit from the pvm console.

```
>pvm mortadelo.hostname
pvm>quit
```

To start the test you will clean the temporary files:

```
>make clean
```

and you can benchmarking the whole calculus:

```
>time make all
```

After the simulations you can see the results with

```
>make plot
```

and **that's all folk!**

```
#!/bin/bash
export EDTDIR=$PWD/EDT/
export TABLES3DDIR=$PWD/EDT/
export PS1="\[\033[0;\w\007 \033[32m\]\u@\h
\[\033[33m\w\033[0m\]\nIIS >"
export PATH=$PWD:$PATH
export IIS=$PWD/iispvm_intel

export PVM_EXPORT=EDTDIR:TABLES3DDIR

cp load GaAs/Se_GaAs
cp load GaAs/Si_GaAs
cp load Si/As_Si_100_8
cp load Si/B_Si_0
cp load Si/B_Si_100_7
cp load Si/Damage_Si
cp load SiC/Al_SiC
cp load SiC/As_SiC

mkdir -p $PWD/Temp

HOSTNAME=`uname -n`
echo "* wd=$PWD/Temp > $HOSTNAME.hostname"
echo "* ep=$PWD >> $HOSTNAME.hostname"
echo $HOSTNAME sp=1000 >> $HOSTNAME.hostname

bash --norc
```

```
#!/bin/bash
# Usage: AddHost host login directory

rsh $1 -l $2 mkdir $3
rcp ../TestPVM.tgz $2@$1:$3
rsh $1 -l $2 "cd $3;tar xzf TestPVM.tgz"

HOSTNAME=`uname -n`

echo $1 sp=1000 ep=$3/TestPVM lo=$2 >> $HOSTNAME.hostname
```