Computación Experimental Usando Herramientas de Software Libre

Fabio A. González O.
Depto. de Ing. de Sistemas e Industrial
Universidad Nacional de Colombia

Problema:

- Debo estudiar el comportamiento de un algoritmo al variar ciertos parámetros, para lo cual debo correr múltiples experimentos, procesar los resultados, crear gráficas y escribir un documento.

Una posible solución:

- Escriba un programa en su lenguaje de preferencia que implemente el algoritmo y el proceso experimental.
- Corra el programa y genere resultados en un archivo plano.
- Cargue este archivo en Excel y cree las gráficas.
- Cree el documento en Word e importe las gráficas desde Excel.

Algunas características de UNIX:

- Creado por y para programadores.
- Una gran caja de herramientas cargada de utilidades desarrolladas por más de 30 años.
- Filosofía:
  - Los programas son herramientas, y por lo tanto deben ser específicos en función pero usables para diferentes propósitos.
  - Los programas están diseñados para trabajar juntos.
- Los programas son independientes de una representación de datos específica y se pueden conectar entre si usando *pipes* (tuberías).
- La mayoría de las herramientas son *libres*.

Algunas herramientas potencialmente útiles para resolver nuestro problema:

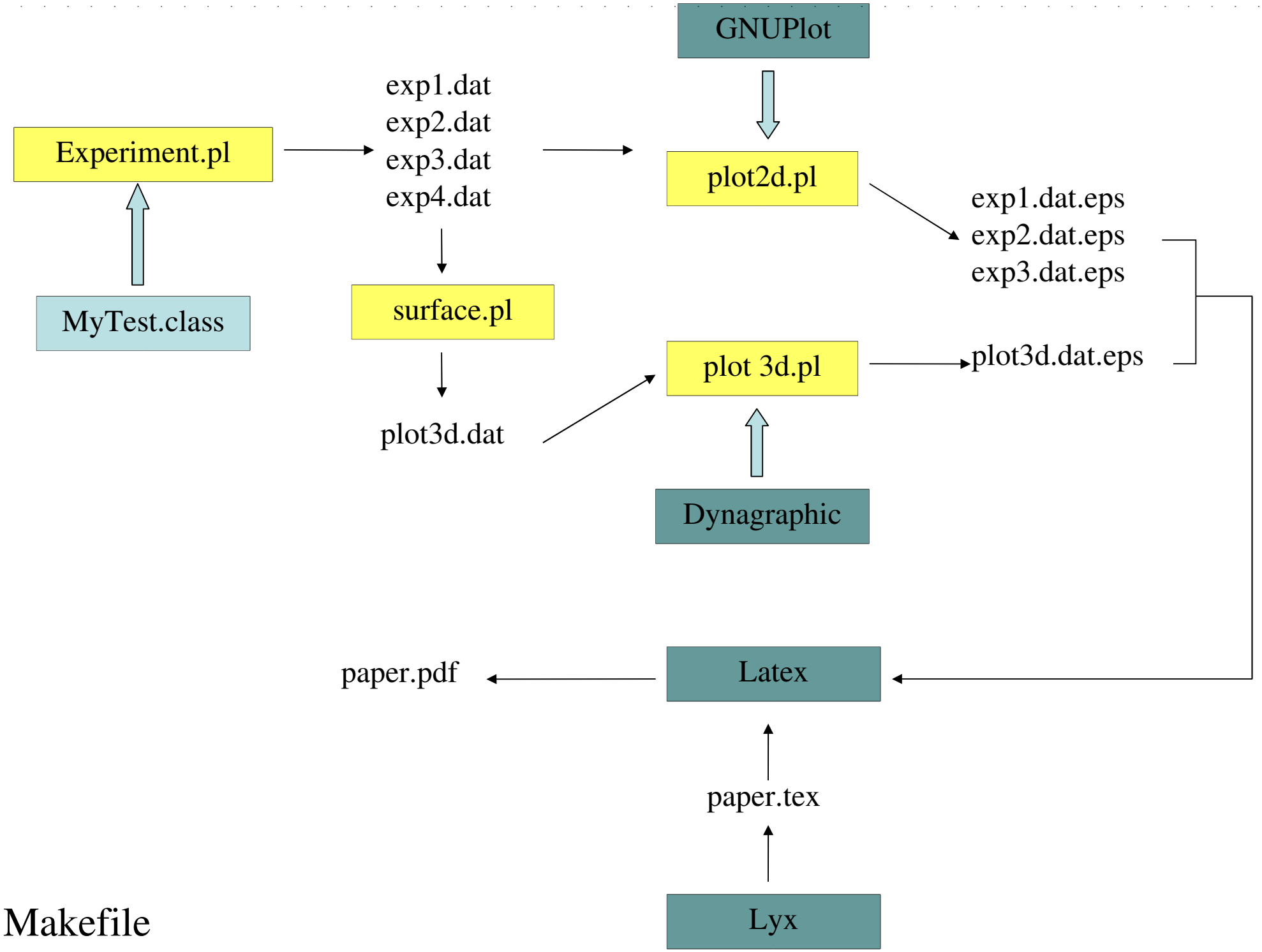
- Lenguajes de script:
  - Shell script, perl, python, etc.
- Graficadores: gnuplot, dynagraph, etc.
- Make: manejo de dependencias entre archivos.
- Lyx y Latex: edición de textos.

Programa a ser estudiado

```
import java.util.*;


public class FreeMem {
  public static void main(String[] args) {
    int numData = Integer.parseInt(args[0]);
    int initSize = Integer.parseInt(args[1]);
    float loadFactor = Float.parseFloat(args[2]);


    Hashtable ht = new Hashtable(initSize,loadFactor);
    for (int i=0 ;i<numData;i++ ) {
        ht.put(new Integer(i),new Integer((int)Math.rint(30000)));
    }
  }
}
```


GNUPlot
exp1.dat
exp2.dat
exp3.dat
exp4.dat
Experiment.pl
plot2d.pl
exp1.dat.eps
exp2.dat.eps
exp3.dat.eps
MyTest.class
surface.pl
plot 3d.pl
plot3d.dat.eps
plot3d.dat
Dynagraphic
paper.pdf
Latex
paper.tex
Lyx

Makefile

Mejoras

- Otro tipo de salidas:
  - Latex2html
- Manejo de versiones:
  - RCS, CVS
- Otros lenguajes de script:
  - Python, Ruby, Javascript

Comentarios finales

- Se pueden correr los experimentos y generar las gráficas cientos de veces (imagine hacer los mismo con Excel!).
- Los módulos son lo suficientemente generales como para reusarlos con pequeñas modificaciones.
- Es portable a cualquier plataforma que permita recompilar los programas (incluido Windows!, usando Cygwin).

experiment.pl

```
#!/usr/bin/perl -w


$maxRuns = $ARGV[0];
$step = $ARGV[1];
$loadFactor = $ARGV[2];

$initSize = 1000;

$classes = "java";

print "#",$loadFactor,"\n";

for($i=0;$i<$maxRuns;$i+=$step){
  $result = `/usr/bin/time -f "\%U \%S" java -cp
   $classes FreeMem $i $initSize $loadFactor 2>&1`;
  @timeOut = split(' ',$result);
  $time = $timeOut[0]+$timeOut[1];
  print $i," ",$time,"\n";
}
```

plot2d.pl

```
#!/usr/bin/perl -w


$file = $ARGV[0];

open(DATA,"<$file");
$linea = <DATA>;
chomp $linea;
$factor = substr($linea,1);
close(DATA);

open(GNPFILE,">/tmp/plot.gnp");
print GNPFILE <<end;

set term post eps
set xlabel "Num data"
set ylabel "Time"
set title "Load Factor = $factor"
set out '$file.eps'
plot "$file" notitle with lines

end

close(GNPFILE);
`gnuplot < /tmp/plot.gnp`;
`rm /tmp/plot.gnp`;
```

plot3d.pl

```
#!/usr/bin/perl -w

$surfFile = $ARGV[0];
open(SCRIPT,">/tmp/surf.dg");

print SCRIPT <<END;
dataplot("$surfFile",title="Hash Table
Performance",scaling=UNCONSTRAINED,axes=BOXED,labels=["Load
Factor","Num Data", "Time"],labelfont=[HELVETICA,BOLD,12],
orientation=[225,70]);
savegraph(cps,"$surfFile.ps");
END

close(SCRIPT);

system("dynagraph < /tmp/surf.dg");
system("rm /tmp/surf.dg");
```

surface.pl

```
#!/usr/bin/perl -w

$result="";
for($i=1;$i<=4;$i++){
  open(DATA,"<exp$i.dat");
  $linea = <DATA>;
  chomp $linea;
  $factor = substr($linea,1);
  $j=0;
  while ($linea=<DATA>){
    chomp $linea;
    $result .= "$factor $linea  ";
    $j++;
  }
  $result .= "\n";
  close(DATA);
  if ($j != 0){
    $numDat=$j;
  }
}

print "SURF 4 $numDat \n",$result;
```

Makefile

```
paper.pdf: paper.dvi
    dvipdf paper.dvi paper.pdf

plots =  exp1.dat.eps exp2.dat.eps exp3.dat.eps plot3d.dat.ps

paper.dvi: paper.tex $(plots)
    latex paper.tex

%.dat.eps : %.dat
    scripts/plot2d.pl $<

plot3d.dat.ps : plot3d.dat
    scripts/plot3d.pl plot3d.dat

plot3d.dat : exp1.dat exp2.dat exp3.dat exp4.dat
    scripts/surface.pl > plot3d.dat

exp1.dat:
    scripts/experiment.pl 30000 3000 0.1 > exp1.dat

exp2.dat:
    scripts/experiment.pl 30000 3000 0.2 > exp2.dat

exp3.dat:
    scripts/experiment.pl 30000 3000 0.4 > exp3.dat

exp4.dat:
    scripts/experiment.pl 30000 3000 4 > exp4.dat

clean:
    rm *.dat *.dat.eps paper.pdf paper.dvi *.ps *.log *.aux
```