`cloudUtil`: Cloud Utilization Visualizations

Christian Panse Ermir Qeli

March 20, 2012

Contents

1 Recent changes and updates

None

2 Introduction

`cloudUtil` is a package for creating comparison plots for Cluster, Grid and Cloud utilization data. Under utilization data we understand collected accounting data measuring the job execution time in the above mentioned environments.

The idea behind this package is to create a single visualization of such data that has the following main features:

- gives an overview over the compute system utilization within a certain time frame
- allows the comparison of job lengths between different platforms giving thus hints on how well the respective job queues function e.g. how efficient the queue of Sun Grid Engine is performing
- allows the integration of replicates within the same visualization
- allows the comparison on both absolute and relative timescales

The functionality of `cloudUtilPlot` function was first used in [1].

3 Data preparation

The package includes sample accounting data for demonstration purposes. These data were collected by comparing the running times of several hundred compute jobs: each one of these jobs performs peptide-spectrum matching in proteomics (data published in [2]).

The fragment below shows a random extract from the dataset provided in the package:

```
> library(cloudUtil)
> data(cloudms2)
> cloudms2[sort(sample(nrow(cloudms2),10)),c(1,5,6,15)]
```

```
       CLOUD BEGIN_PREPROCESS END_PREPROCESS   id
184    EC2_1       1263526345     1263526348 1678
904    FGCZ1       1263120848     1263120858 1454
1322   EC2_1       1263520374     1263520376 1402
2481    UZH2       1263449814     1263449816 1794
2970   EC2_2       1263542314     1263542326  442
4633   EC2_2       1263586776     1263586776 1822
4777    UZH1       1261621056     1261621085  307
6836    UZH1       1261637735     1261637743  908
9521   FGCZ2       1263436557     1263436568 1030
10517  EC2_1       1263479571     1263479607   26
```

The attributes of interest are `CLOUD`, `BEGIN_PREPROCESS`, `END_PREPROCESS`, and `id`. Additionally, it is also possible to use accounting data collected from other sources e.g. Sun Grid Engine accounting data [3].

4 Analysis

The code extract below creates a plot of the data shown in Section 3:

```
> hist(cloudms2$END_PREPROCESS - cloudms2$BEGIN_PREPROCESS,100)
> ##
> boxplot((cloudms2$END_PROCESS-cloudms2$BEGIN_PROCESS)/3600~cloudms2$CLOUD,
+     main="process time",
+     ylab="time [hours]")
> ##
> throughput<-cloudms2$MZXMLFILESIZE*10^-6/
+ (cloudms2$END_COPYINPUT-cloudms2$BEGIN_COPYINPUT)
> boxplot(throughput~cloudms2$CLOUD,
+     main="copy input network throughput",
+     ylab="MBytes/s")
> ##
>
> cloudUtilPlot(begin=cloudms2$BEGIN_PROCESS,
+     end=cloudms2$END_PROCESS,
+     id=cloudms2$id,
+     group=cloudms2$CLOUD)
```

Transparency through alpha blending allows furthermore to compare several plots with each other. An example is given in the code fragment below:

```
> #green
> col.amazon<-rgb(0.1,0.8,0.1,alpha=0.2)
> col.amazon2<-rgb(0.1,0.8,0.1,alpha=0.2)
> #blue
> col.fgcz<-rgb(0.1,0.1,0.8,alpha=0.2)
> col.fgcz2<-rgb(0.1,0.1,0.5,alpha=0.2)
> #red
> col.uzh<-rgb(0.8,0.1,0.1,alpha=0.2)
> col.uzh2<-rgb(0.5,0.1,0.1,alpha=0.2)
> cm<-c(col.amazon, col.amazon2, col.fgcz, col.fgcz2, col.uzh, col.uzh2)
> jpeg("cloudms2Fig.jpg", 640, 640)
> op<-par(mfrow=c(2,1))
> cloudUtilPlot(begin=cloudms2$BEGIN_PROCESS,
+     end=cloudms2$END_PROCESS,
+     id=cloudms2$id,
+     group=cloudms2$CLOUD,
+     colormap=cm,
+     normalize=FALSE,
+     plotConcurrent=TRUE);
> cloudUtilPlot(begin=cloudms2$BEGIN_PROCESS,
+     end=cloudms2$END_PROCESS,
+     id=cloudms2$id,
+     group=cloudms2$CLOUD,
```

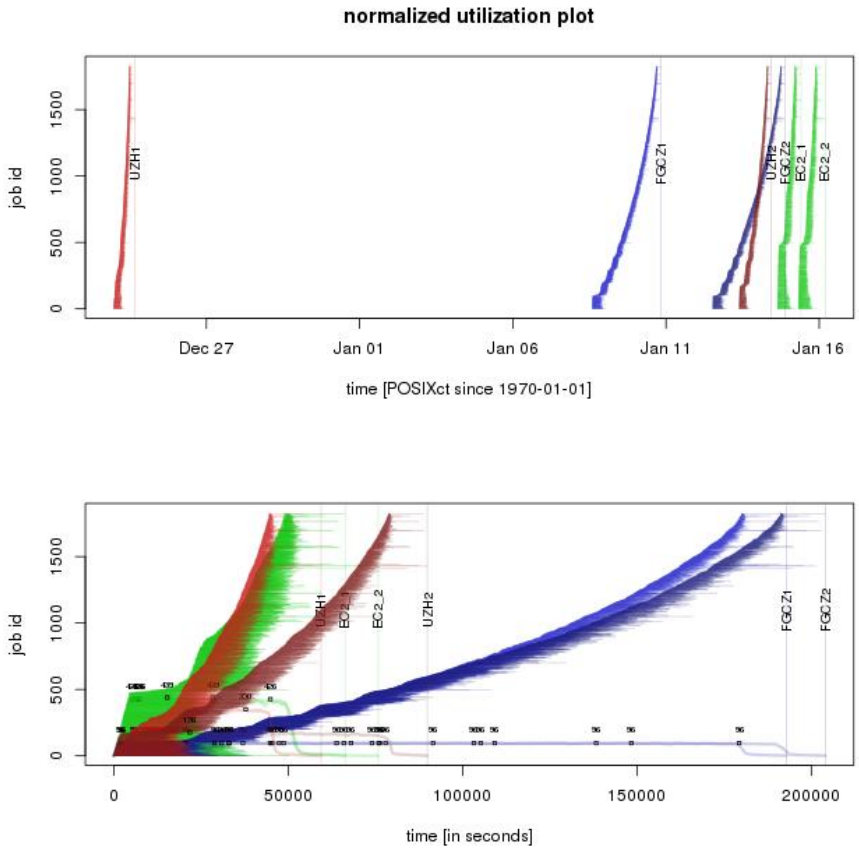

Figure 1: `cloudUtilPlot` visualization for the `cloudms2` data set. On the graphics each horizontal line indicates the start and the end of one single job. Color is used for classifying the different groups. On the upper plot the time of each group was not normalized. The visualization on the bottom on the other side uses normalized time scales whichs help to compare the compute systems. Tranparent colors were used to dial with the overplotting. The solid lines on the bottom plot show the total number of concurrently running jobs. The squares on the solid lines indicate the maxima on the respective system. The user can make use of all R graphic devices.

```
+     colormap=cm,
+     normalize=TRUE,
+     plotConcurrent=TRUE,
+     plotConcurrentMax=TRUE)
> dev.off()

pdf
  2
```

The output of the above listed R session is shown in Figure 1.

References

[1] Aleksandar Markovic, Investigation of economical and practical aspects of commercial cloud computing for Life Sciences, March 2010. (diplomathesis)

[2] Erich Brunner et al., A high-quality catalog of the Drosophila melanogaster proteome., Nat Biotechnol. 2007 May;25(5):576-83. Epub 2007 Apr 22., (pubmed ID:17450130).

[3] `https://sourceforge.net/projects/gridscheduler/`, March 2012.