`cloudUtil`: Cloud Utilization Visualizations

Christian Panse Ermir Qeli

June 9, 2016

Contents

1 Recent changes and updates

'vignettes' directory has been migrated.

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 [3].

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 [1]).

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
1348  FGCZ2       1263480345     1263480348 1630
2085  EC2_1       1263510160     1263510169  969
3747   UZH1       1261633035     1261633048  678
3847   UZH2       1263408520     1263408549  579
3889   UZH2       1263421226     1263421248  865
4642  EC2_1       1263508062     1263508069  912
6744   UZH1       1261646787     1261646792 1390
7531  FGCZ1       1263001139     1263001162  280
10760  UZH1       1261645841     1261645847 1330
10872 FGCZ2       1263297904     1263297913   96
```

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 [2].

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,
```

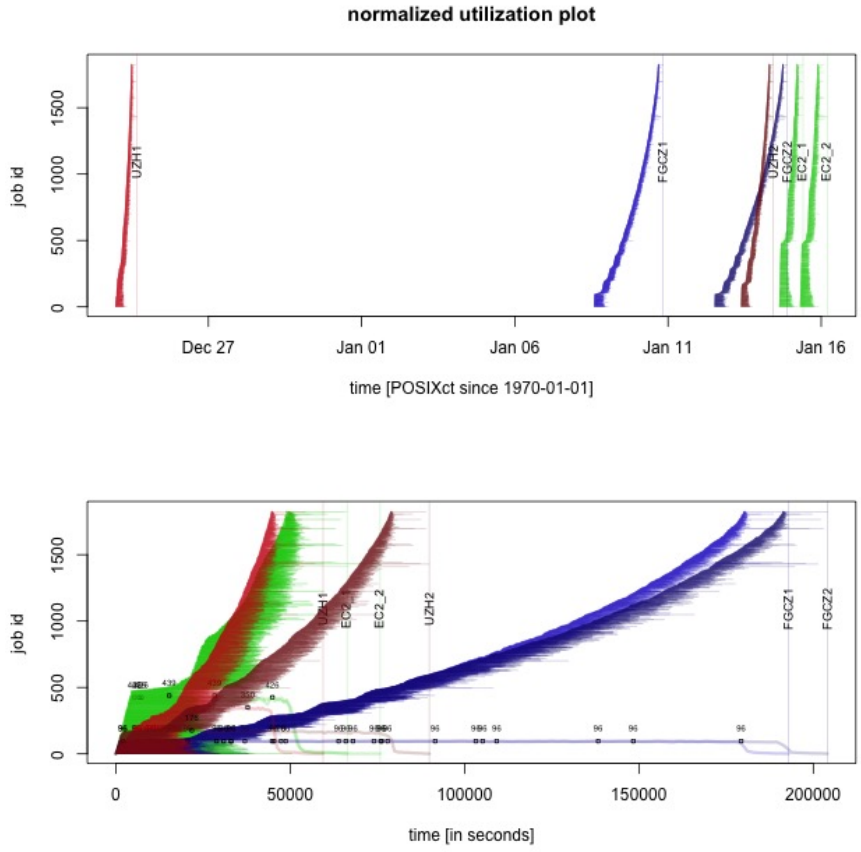

Figure 1: `cloudUtilPlot` visualization for the `cloudms2` data set. On the graphics each horizontal line indicates the start and the end of one single job. Color is used for classifying the different groups. On the upper plot the time of each group was not normalized. The visualization on the bottom 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] E. Brunner, C. H. Ahrens, S. Mohanty, H. Baetschmann, S. Loevenich, F. Potthast, E. W. Deutsch, C. Panse, U. de Lichtenberg, O. Rinner, H. Lee, P. G. Pedrioli, J. Malmstrom, K. Koehler, S. Schrimpf, J. Krijgsveld, F. Kregenow, A. J. Heck, E. Hafen, R. Schlapbach, and R. Aebersold. A high-quality catalog of the Drosophila melanogaster proteome. *Nat. Biotechnol.*, 25(5):576–583, May 2007. [DOI:10.1038/nbt1300] [PubMed:17450130].

[2] Rayson Ho and Ron Chen. Open grid scheduler. `https://sourceforge.net/projects/gridscheduler`, 2013.

[3] Aleksandar Markovic. Investigation of economical and practical aspects of commercial cloud computing for life sciences. Master's thesis, 2010.