

# Package: cloudUtil (via r-universe)

August 18, 2024

**Type** Package

**Title** Cloud Utilization Plots

**Version** 0.1.12

**Author** Christian Panse <cp@fgcz.ethz.ch>, Ermir Qeli  
<ermir.qeli@gmail.com>

**Maintainer** Christian Panse <cp@fgcz.ethz.ch>

**Depends** R (>= 2.11.0), methods

**Description** Provides means of plots for comparing utilization data of  
compute systems.

**License** GPL-2

**Collate** cloudUtilPlot.R

**URL** <https://cran.r-project.org/package=cloudUtil>

**NeedsCompilation** no

**Date/Publication** 2016-06-09 15:19:22

**Repository** <https://cpanse.r-universe.dev>

**RemoteUrl** <https://github.com/cran/cloudUtil>

**RemoteRef** HEAD

**RemoteSha** 33a440a0d49b6f50b3c07bb0ca364287daa75ee2

## Contents

cloudms2 . . . . .	2
cloudUtilPlot . . . . .	2

<b>Index</b>	<b>5</b>
--------------	----------

---

`cloudms2`*Benchmark data set for cloudUtil*

---

**Description**

The data set provides the measurements of benchmark using three compute infrastructures namely uzh.ch Schroedinger, Amazon EC2 (region us-east.), and the FGCZ internal compute server. The number crunching was done by doing tandem mass spectra peptide sequence assignments of a large scale proteomics Drosophila experiment similar to the data processing published here [E. Brunner, et. al., NBT, 2007] having approx. 1800 (LC)-MS/MS runs, 8474960 fragment ion spectra (tandem MS or MS2), identifying 498000 redundant and 72281 distinct peptides, and 9124 proteins. The data volume is approx. 0.3TB splitted into ~1800 jobs.

**Format**

A data frame with 10969 rows and 15 variables

**References**

<http://fgcz-transfer.uzh.ch/~cpanse/2011-10-24--pCloud.pdf>

<http://www.nature.com/nbt/journal/v25/n5/abs/nbt1300.html>

---

`cloudUtilPlot`*Cloud Utilization Plot*

---

**Description**

On the utilization graphic each horizontal line indicates the start and the end of one job. Color groups different runs.

**Usage**

```
cloudUtilPlot(group, begin, end, id,  
              colormap,  
              normalize,  
              plotConcurrent,  
              plotConcurrentMax,  
              main)
```

## Arguments

begin	a time series of the starting events; the time format is unix time stamp (seconds since Jan 01 1970).
end	a time series of the corresponding ending events.
id	identifiyer of each event.
group	group of each event.
main	an overall title for the plot.
colormap	color vector for the data. default is the rainbow colormap having as much colors as we have groups.
normalize	normalizes the time in a way that the groups are relative to each other; default is set to TRUE.
plotConcurrent	plot concurrent running tasks as solid lines; default is set to TRUE.
plotConcurrentMax	plot maximum of concurrent running tasks; default is set to FALSE.

## Details

The cloudUtilPlot function requires the four arguments begin, end, group, and id. All other arguments are optional.

## Author(s)

Christian Panse

## References

Tyanko Aleksiev, Simon Barkow, Peter Kunszt, Sergio Maffioletti, Riccardo Murri, Christian Panse (2013), VM-MAD: a cloud/cluster software for service-oriented academic environments, <https://arxiv.org/abs/1302.2529>.

## Examples

```
data(cloudms2)

#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.fgc2<-rgb(0.1,0.1,0.8,alpha=0.2)
col.fgc22<-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.fgc2, col.fgc22, col.uzh, col.uzh2)
```

```
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,  
              colormap=cm,  
              normalize=TRUE,  
              plotConcurrent=TRUE)
```

# Index

## \* datasets

cloudms2, [2](#)

cloudms2, [2](#)

cloudUtil (cloudUtilPlot), [2](#)

cloudUtilPlot, [2](#)