

# JoSy Orchestration

Pourquoi les dinosaures sont toujours là,  
même si l'on n'aime pas les piafs

# Qui suis-je ?

On Mon, 2022-03-07 at 11:42 +0100, Sylvain Maurin wrote:  
Subject : Proposition - Pourquoi les dinosaures sont toujours là,  
même si l'on n'aime pas les piafs

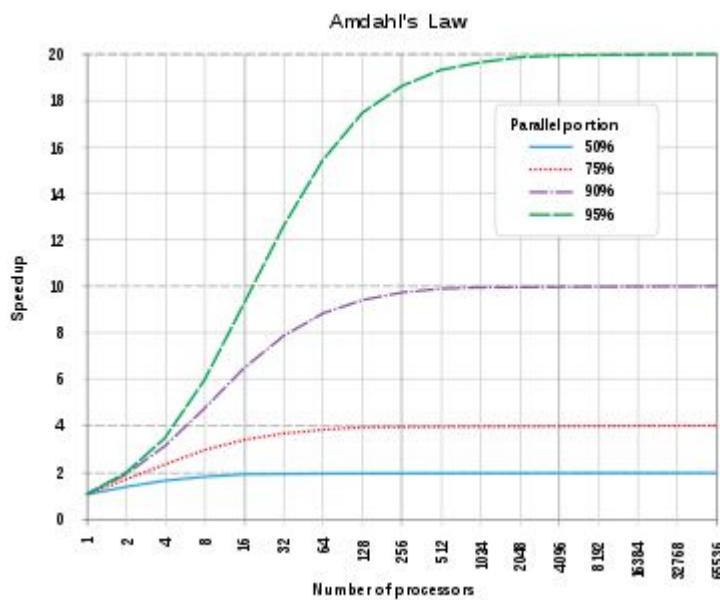
[...]

> Sans dire que c'est parfait, je défendrai mon expérience personnelle en  
> rappelant qu'il y a un certain nombre d'arguments et de contextes qui à  
> mon idée, impliquent de savoir faire le choix d'éviter les frais d'un  
> outils d'orchestration :  
> - La gestion d'une flotte d'échantillons,  
> - Le danger d'une maintenance par point unique d'accès, en particulier  
> dans les cas où l'adoption de l'outil est plus long que les bonnes  
> pratiques d'usage,  
> - La difficulté de reproductibilité et de réPLICATION des expériences  
> d'ASR dans le cas d'équipe de laboratoire unipersonnelle,  
> - Le risque des "recettes" dans le cas où les ingrédients ne sont pas  
exactement ceux attendus et que le cuistot n'est pas bon,

[...]

# Gene Amdahl

$$S_{\text{latence}}(s) = \frac{1}{1 - p + \frac{p}{s}}$$



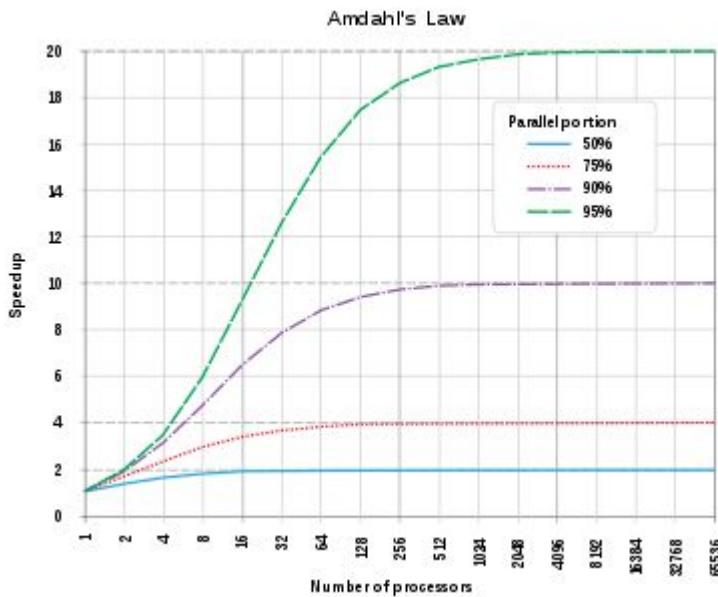
La loi d'Amdahl est souvent utilisée en calcul parallèle pour prédire l'accélération théorique lors de l'utilisation de plusieurs processeurs. Par exemple, si un programme a besoin de 20 heures d'exécution sur un processeur uni-coeur et qu'une partie du programme qui requiert une heure d'exécution ne peut pas être parallélisée, même si les 19 heures ( $p = 95\%$ ) d'exécution restantes peuvent être parallélisées, quel que soit le nombre de processeurs utilisés pour l'exécution parallèle du programme, le temps d'exécution minimal ne pourra passer sous cette heure critique. Ainsi, l'accélération théorique est limitée au plus à 20 ( $1/(1-p) = 20$ ).

On en déduit deux règles : premièrement, lors de l'écriture d'un programme parallèle, il faut limiter autant que possible la partie serielle ; deuxièmement, un ordinateur parallèle doit être un excellent ordinateur sériel pour traiter le plus rapidement possible la partie serielle.

[https://en.wikipedia.org/wiki/Amdahl's\\_law](https://en.wikipedia.org/wiki/Amdahl's_law)

# Gene Amdahl lu par l'AdminSys

$$S_{\text{latence}}(s) = \frac{1}{1 - p + \frac{p}{s}}$$



La loi d'Amdahl est souvent utilisée en administration système pour prédire l'**accélération** théorique lors de l'utilisation d'un **système d'orchestration**. Par exemple, si une tache d'administration a besoin de 20 heures de travail d'AdminSys sur un processus automatisable et qu'une partie de la tache requiert une heure de travail qui ne peut pas être automatisée, même si les 19 heures ( $p = 95\%$ ) de taches restantes peuvent être automatisées, quel que soit le nombre d'astuces d'automatisation utilisées par l'adminsyst, le temps d'exécution minimal ne pourra passer sous cette heure critique. Ainsi, l'accélération théorique est limitée au plus à 20 ( $1/(1 - p) = 20$ ).

On en déduit deux règles : premièrement, lors de l'écriture d'un automatisme d'orchestration, il faut limiter autant que possible la partie humaine ; deuxièmement, une infrastructure « orchestrable » doit être pourvue d'un excellent AdminSys pour traiter le plus rapidement possible la partie non automatisable.

[https://en.wikipedia.org/wiki/Amdahl's\\_law](https://en.wikipedia.org/wiki/Amdahl's_law)

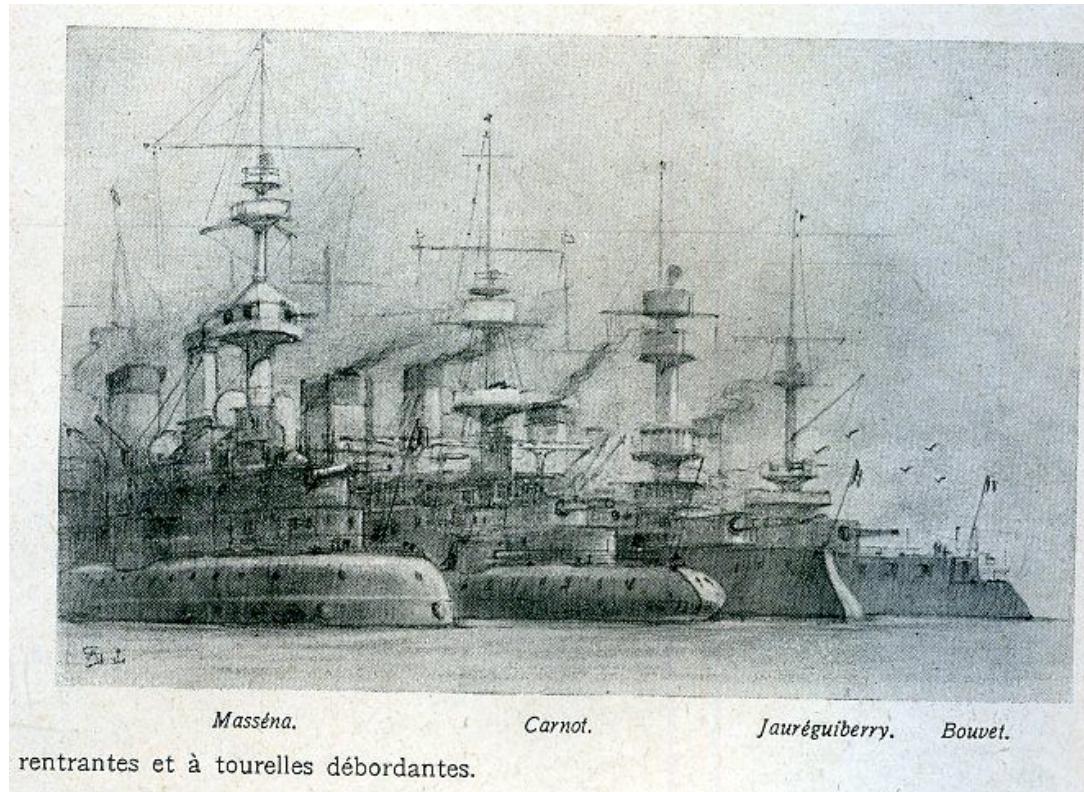
# K.I.S.S.

<http://people.apache.org/~fhanik/kiss.html>

- Be Humble, don't think of yourself as a super genius, this is your first mistake
- By being humble, you will eventually achieve super genius status =), and even if you don't, who cares! **your code is stupid simple, so you don't have to be a genius to work with it.**
- Break down your tasks into **sub tasks** that you think should take no longer than **4-12 hours to code**
- Break down your problems into many small problems. Each problem should be able to be solved within one or a very few classes
- **Keep your methods small, each method should never be more than 30-40 lines.** Each method should only solve one little problem, not many uses cases
- If you have a **lot of conditions in your method, break these out** into smaller methods.
- Not only will this be easier to read and maintain, but you will find bugs a lot faster.
- You will learn to love Right Click+Refactor in your editor.
- Keep your classes small, same methodology applies here as we described for methods.
- **Solve the problem, then code it.** Not the other way around
- Many developers solve their problem while they are coding, and there is nothing wrong doing that. As a matter of fact, you can do that and still adhere to the above statement.
- If you have the ability to mentally **break down things into very small pieces**, then by all means, do that while you are coding. But don't be afraid of refactor your code over and over and over again. Its the end result that counts, and number of lines is not a measurement, unless you measure that fewer is better of course.
- **Don't be afraid to throw away code.** Refactoring and recoding are two very important areas. As you come across requirements that didn't exist, or you weren't aware of when you wrote the code to begin with you might be able to solve the old and the new problems with an even better solution.
- If you had followed the advice above, the amount of code to rewrite would have been minimal, and if you hadn't followed the advice above, then the code should probably be rewritten anyway.
- And for all other scenarios, **try to keep it as simple as possible**, this is the hardest behavior pattern to apply to, but once you have it, you'll look back and will say, I can't imagine how I was doing work before.

<http://people.apache.org/~fhanik/kiss.html>

# Flotte d'échantillon



[https://fr.wikipedia.org/wiki/Flotte\\_d'échantillons](https://fr.wikipedia.org/wiki/Flotte_d'échantillons)

# Parc de laboratoire : homogénéité & hétérogénéité

# Temps d'apprentissage des outils

Utilisez Ansible pour automatiser vos tâches de configuration

6 heures Moyenne

Licence CC BY SA

Mis à jour le 14/06/2022

Créé par

OpenClassrooms, Leading E-Learning Platform in Europe

Vous êtes administrateur système et vous en avez assez des **tâches répétitives**? Vous souhaitez mettre en place des pratiques DevOps d'**automatisation** dans vos déploiements et vos configurations? Ce cours est fait pour vous!

<https://openclassrooms.com/fr/courses/2035796-utilisez-ansible-pour-automatiser-vos-taches-de-configuration>

*« Every time someone logs onto a system by hand, they jeopardize everyone's understanding of the system. »*

—Mark Burgess, author of CFEngine

Learning CFEngine 3, by Diego Zamboni  
ISBN: 9781449312206

*Three Rings for the Elven-kings under the sky,  
Seven for the Dwarf-lords in their halls of stone,  
Nine for Mortal Men doomed to die,  
One for the Dark Lord on his dark throne,  
In the Land of Mordor where the Shadows lie,  
One Ring to rule them all, one Ring to find them,  
One Ring to bring them all and in the darkness bind them  
In the Land of Mordor where the Shadows lie*

— J.R.R. Tolkien, The Lord of the Rings, Epigraph

USENIX Summer Conference  
June 11-15, 1990  
Anaheim, California

Don Libes – National Institute of  
Standards and Technology

# expect: Curing Those Uncontrollable Fits of Interaction

## ABSTRACT

UNIX programs used to be designed so that they could be connected with pipes created by a shell. This paradigm is insufficient when dealing with many modern programs that *demand* to be used interactively.

**expect** is a program designed to control interactive programs. **expect** reads a script that resembles the dialogue itself but which may include multiple paths through it. Scripts include:

- send/expect sequences – **expect** patterns can include regular expressions.
- high-level language – Control flow (**if/then/else**, **while**, etc.) allows different actions on different inputs, along with procedure definition, built-in expression evaluation, and execution of arbitrary UNIX programs.
- job control – Multiple programs can be controlled at the same time.
- user interaction – Control can be passed from scripted to interactive mode and vice versa at any time. The user can also be treated as an I/O source/sink.

**expect** successfully deals with interactive programs. It also solves several other large classes of problems which UNIX shells do not.

# Le diskless : 1 NFSRoot pour tous !

## ■ Des postes clients

Quatre-vingt dix pour cent des postes client sont de type PC : du 486 au Pentium 400 (une politique de renouvellement en cours tend à supprimer toutes les machines ne passant pas l'an 2000 (surtout des PC à base de Intel 486) et tous les mac ne supportant pas Mac OS 8.5).

Ainsi :

- Sur le Réseau Recherche et Administration :  
Seul un système acceptant un mode super-administrateur (Unix, NT...) sauf pour les Mac (Mac OS 8.5) en attendant Mac OS X, est accepté pour les postes à raccorder au réseau sécurisé.
- Sur le réseau étudiant :  
Plus de souplesse que sur le Réseau Recherche/Administration avec Windows 95 en mode pseudo-diskless.

En effet dans le cas des étudiants et des postes en libre service les solutions **diskless** W95 ou linux avec un temps de boot suffisamment court associées à une messagerie de type WEBMAIL sont très intéressantes car elles permettent d'assurer une disponibilité élevée des salles libre service mais aussi un redémarrage sur incident en cours de TP très bref.

La très grande majorité des postes sont bi-systèmes (Windows/Linux) et utilise une boot-prom PXE présente sur la carte réseau ou une disquette d'émulation en cas de défaut voire l'utilisation du GHOST Multicast.

Le partage des ressources est assuré via NFS, NIS et SAMBA 2.0 côté serveur. Le client SMB utilisé pour les Mac est DAVE.

En cours : Nous n'avons jamais été complètement satisfaits de NIS et en réalité nous avons synchronisé par nous même les serveurs à partir d'un serveur maître sous OpenStep (Mac OS X serveur) utilisant NetInfo (que l'on peut assimiler à NDS de Novell comme annuaire maître). Notre objectif est de mettre en place cette année un annuaire global compatible LDAP. Le choix n'est pas encore fait.

<https://1999.jres.org/articles/coris-adm-01-finalps.pdf>

# Le script Kleenex©

```
bombadil:/root# cat /usr/local/lib/host-linux
zmani
n18h201
n18h202
n18h203
n18h204
...
#n18h205 aka thorin
n18h206
#n18h207 shutdown
n18h208
n18h209
n18h210
n18h211
#n18h212 shutdown
n18h213
n18h214
n18h215
N18h216
...
n18h217
n18h218
n18h221
n19h2
n19h3
n19h4
n19h5
n19h6b
n19h7
#n19h8
#n19h9
#n19h10
n19h11
n19h17
#n19h19
#n19h21
#n19h35
...
#n19h130
#n19h132
#n19h133
#n19h134
#n19h135
#n19h137
#n19h176
#cheylus
frodon
#gimli
legolas
saeros
vmecas216
vmecas217
vmecas218
```

```

$ cat /usr/local/sbin/distribute-debian-grep-password.sh
#!/bin/bash

GREPREG=$1

if (( $DEBUG ))
then
#_OPT=""
set -x
fi

IND=0

for DESTINATION in `cat /usr/local/lib/host-linux | egrep -v "^[[:space:]]*#"` `cat /usr/local/lib/shared-srv-linux | egrep -v "^[[:space:]]*#"`
do
export PARENTPID=$$
export DESTINATION
export IND=$((IND+1))
(
LOG=
echo -n -e "== $DESTINATION ==\n" ; su - save -c "ssh $DESTINATION -l root 'egrep $GREPREG /etc/passwd' " 2>&1
echo "$LOG" >/tmp/$PARENTPID.$IND
) &
done
wait
for ((I=1;I<=$IND;I++))
do
cat /tmp/$PARENTPID.$I
rm /tmp/$PARENTPID.$I
done

```

```
bombadil:/root# ls /usr/local/sbin/distribute-debian-*
/usr/local/sbin/distribute-debian-NetWorkManager_nmcli_dev_status.sh
/usr/local/sbin/distribute-debian-apt-apt.conf-from-bombadil.sh
/usr/local/sbin/distribute-debian-apt-source.list-from-bombadil.sh
/usr/local/sbin/distribute-debian-aptclean.sh
/usr/local/sbin/distribute-debian-aptitude.sh
/usr/local/sbin/distribute-debian-bombadil_rootAuthorized_keys2.sh
/usr/local/sbin/distribute-debian-df/usr-local-matlab.sh
/usr/local/sbin/distribute-debian-df.sh
/usr/local/sbin/distribute-debian-download-upgrade.sh
/usr/local/sbin/distribute-debian-file-modify-syslog-ng.conf
/usr/local/sbin/distribute-debian-fsl-fsl.conf-from-bombadil.sh
/usr/local/sbin/distribute-debian-fsl-fsl_setup.bash-from-bombadil.sh
/usr/local/sbin/distribute-debian-grep-password.sh
/usr/local/sbin/distribute-debian-halt-all.sh
/usr/local/sbin/distribute-debian-halt-host-linux.sh
/usr/local/sbin/distribute-debian-halt-shared-srv.sh
/usr/local/sbin/distribute-debian-host-linux-nrpe-install.sh
/usr/local/sbin/distribute-debian-install-package-host-linux.pl
/usr/local/sbin/distribute-debian-install-package-host-linux.sh
/usr/local/sbin/distribute-debian-install-package-shared-srv-linux.pl
/usr/local/sbin/distribute-debian-install-package.pl
/usr/local/sbin/distribute-debian-install-package.sh
/usr/local/sbin/distribute-debian-invoke-rc.d_syslog-ng_restart.sh
/usr/local/sbin/distribute-debian-loghost_conf.sh
/usr/local/sbin/distribute-debian-motd.sh
```

```
/usr/local/sbin/distribute-debian-ntp-date-sync.sh
/usr/local/sbin/distribute-debian-ntp-date-sync.sh.1
/usr/local/sbin/distribute-debian-ntpdate-sync.sh
/usr/local/sbin/distribute-debian-ntpdc-peers.sh
/usr/local/sbin/distribute-debian-purge-package-host-linux.sh
/usr/local/sbin/distribute-debian-purge-package-shared-srv-linux.pl
/usr/local/sbin/distribute-debian-purge-package-shared-srv-linux.sh
/usr/local/sbin/distribute-debian-purge-package.sh
/usr/local/sbin/distribute-debian-reboot-shared-srv-linux.sh
/usr/local/sbin/distribute-debian-reboot.sh
/usr/local/sbin/distribute-debian-reinstall-package-host-linux.sh
/usr/local/sbin/distribute-debian-root_.bashrcf-from-bombadil.sh
/usr/local/sbin/distribute-debian-screen-ls.sh
/usr/local/sbin/distribute-debian-ssh-vulnkey.sh
/usr/local/sbin/distribute-debian-testROfs.sh
/usr/local/sbin/distribute-debian-touch_etc_environnement.sh
/usr/local/sbin/distribute-debian-uname.sh
/usr/local/sbin/distribute-debian-update.sh
/usr/local/sbin/distribute-debian-upgrade-host-linux.sh
/usr/local/sbin/distribute-debian-upgrade-shared-srv-linux.sh
/usr/local/sbin/distribute-debian-upgrade-trivial.sh
/usr/local/sbin/distribute-debian-upgrade.sh
/usr/local/sbin/distribute-debian-uptime.sh
/usr/local/sbin/distribute-debian-users-host-linux.sh
/usr/local/sbin/distribute-debian-version.sh
```

```
bombadil:/root# cat /usr/local/*/PowerEdge6220Lames
n19h2
n19h3
n19h4
n19h5
n19h132
n19h133
n19h134
n19h135
bombadil:/root# grep PowerEdge6220Lames /usr/local/sbin/*
/usr/local/sbin/distribute-install-PowerEdge.sh: `cat /usr/local/lib/PowerEdge6220Lames | egrep -v "^[[:space:]]*#"`
```

```

bombadil:/root# cat /usr/local/sbin/distribute-install-PowerEdge.sh
#!/bin/bash
for DESTINATION in \
`cat /usr/local/lib/PowerEdge6220Lames | egrep -v "^[[:space:]]*#"`\
do
(
    LOG=
    echo ""
    echo ""
    echo "*** $DESTINATION ****"
    ping -w 5 -c 3 -q $DESTINATION >/dev/null 2>&1
    if [[ $? == 0 ]]
    then
        su - save --command="ssh $DESTINATION -l root 'pkill rsync ' " 2>&1
        su - save --command="ssh $DESTINATION -l root 'aptitude update ; aptitude -y upgrade ; aptitude -y install rsync screen
logcheck logwatch screen' " 2>&1
        su - save --command="ssh $DESTINATION -l root ' umount /dev/sdb3 ; mount ; rm -vr /mnt/*' " 2>&1
        su - save --command="ssh $DESTINATION -l root 'mkdir -p /mnt/backup-online ; mount /dev/sdb3 /mnt/backup-online' " 2>&1
        su - save --command="ssh $DESTINATION -l root 'dd if=/dev/zero of=/mnt/backup-online/ZERO_FILE_AFAC count=10240 bs=1024k ;
' " 2>&1
        su - save --command="ssh $DESTINATION -l root 'rsync -aH --delete --one-file-system /. /mnt/backup-online/. ' " 2>&1
        su - save --command="ssh $DESTINATION -l root 'echo $DESTINATION >/root/sda ; echo $DESTINATION >/mnt/root/sdb' " 2>&1
    else
        echo "No answer in ICMP ?"
    fi
    echo "$LOG"
) &
done
wait
echo

```

```

#!/bin/bash
for DESTINATION in \
`cat /usr/local/lib/PowerEdge-Buster | egrep -v "^[[:space:]]*#"`
do
echo ""
echo "*** $DESTINATION ***"
ping -w 5 -c 3 -q $DESTINATION >/dev/null 2>&1
if [[ $? == 0 ]]
then
    su -l save -c "ssh $DESTINATION -t root 'cat /proc/partitions' " 2>&1
    su -l save -c "ssh $DESTINATION -t root 'blkid' " 2>&1
    su -l save -c "ssh $DESTINATION -t root 'chmod 1777 /scratch' " 2>&1
else
    echo "No answer in ICMP ?"
fi
done
wait
echo

```

```
bombadil:/root# ls /usr/local/sbin/distribute-install-PowerEdge-stretch.sh*
/usr/local/sbin/distribute-install-PowerEdge-stretch.sh          /usr/local/sbin/distribute-install-PowerEdge-stretch.sh.3
/usr/local/sbin/distribute-install-PowerEdge-stretch.sh.1        /usr/local/sbin/distribute-install-PowerEdge-stretch.sh.4
/usr/local/sbin/distribute-install-PowerEdge-stretch.sh.10       /usr/local/sbin/distribute-install-PowerEdge-stretch.sh.5
/usr/local/sbin/distribute-install-PowerEdge-stretch.sh.11       /usr/local/sbin/distribute-install-PowerEdge-stretch.sh.7
/usr/local/sbin/distribute-install-PowerEdge-stretch.sh.12       /usr/local/sbin/distribute-install-PowerEdge-stretch.sh.8
/usr/local/sbin/distribute-install-PowerEdge-stretch.sh.2        /usr/local/sbin/distribute-install-PowerEdge-stretch.sh.9
```

```

bombadil:/root# cat /usr/local/sbin/distribute-install-PowerEdge-stretch.sh.3
#!/bin/bash
for DESTINATION in \
    `cat /usr/local/lib/PowerEdge-Strech | egrep -v "^[[:space:]]*#"`"
do
echo ""
echo ""
echo "**** $DESTINATION ****"
ping -w 5 -c 3 -q $DESTINATION >/dev/null 2>&1
if [[ $? == 0 ]]
then
    su - save --command="ssh $DESTINATION -l root -t 'dpkg --configure -a ; apt-get update ; apt-get -y upgrade ; apt-get -y install rsync screen logcheck logwatch ''"
    for MYDPKG in \
        task-ssh-server vim \
        nfs-common nfs-kernel-server nfs4-acl-tools nfstrace nfstrace-doc nfswatch \
        dirmngr sudo task-xfce-desktop
    do
        echo "***** $MYDPKG *****"
        su - save --command="ssh $DESTINATION -l root -t 'echo ; echo ${MYDPKG} ; date ; apt-get -y autoremove ; apt-get -y install ${MYDPKG}""
    done
#
#    echo -en "\n\nnn19h13.isc.cnrs.fr:/cncnfs0/$DESTINATION /home nfs fsc 0 0\n" | \
#    su - save --command="ssh $DESTINATION -l root 'cat >/etc/fstab'"
#
#    echo -en "\n\nnn19h13.isc.cnrs.fr:/cncnfs0/common /home/common nfs fsc 0 0\n" | \
#    su - save --command="ssh $DESTINATION -l root 'cat >/etc/fstab'"
#
su - save --command="ssh $DESTINATION -l root 'cat /etc/fstab'"

for MYDPKG in \
    task-french task-french-desktop task-gnome-desktop task-chinese-s \
    task-chinese-s-desktop task-chinese-t task-chinese-t-desktop task-english x11vnc xvfb \
    tigervnc-common tightvncserver science-neuroscience-cognitive science-neuroscience-modeling \
    science-imageanalysis science-linguistics science-statistics gnome xfce4 \
    xserver-xorg-video-dummy linux-source
do
echo "***** ${MYDPKG} *****"
su - save --command="ssh $DESTINATION -l root -t 'echo ; echo ${MYDPKG} ; date ; apt-get -y autoremove ; apt-get -y install ${MYDPKG}""
done

su - save --command="ssh $DESTINATION -l root -t 'apt-get -y build-dep linux-image-amd64'"
su - save --command="ssh $DESTINATION -l root -t 'wget -O http://neuro.debian.net/lists/stretch.de-md.full | sudo tee /etc/apt/sources.list.d/neurodebian.sources.list'"
su - save --command="ssh $DESTINATION -l root -t 'apt-key adv --recv-keys --keyserver hkp://pool.sks-keyservers.net:80 0xA5D32F012649A5A9'"
su - save --command="ssh $DESTINATION -l root -t 'dpkg --configure -a ; apt-get update ; apt-get -y upgrade ;''"
for MYDPKG in \
    caret-data afni afni-common afnix-doc afnix afni-atlases crrun-tools connectome-workbench \
    convert3d eeglab11-sampleddata fsl-complete fsl-harvard-oxford-atlases \
    fsl-harvard-oxford-cortical-lateralized-atlas fsl-jhu-dti-whitematter-atlas \
    fsl-juelich-histological-atlas fsl-melview fsl-mni152-templates \
    fsl-mni-structural-atlas fsl-neurosynth-atlas fslview \
    fslview-doc fsl-neurosynth-top100-atlas fsl-oxford-striatal-atlas \
    fsl-oxford-thalamic-connectivity-atlas fsl-possum-data fsl-resting-connectivity-parcellation-atlases \
    fsl-subthalamic-nucleus-atlas fsl-talairach-daemon-atlas fsl-harvard-oxford-atlases \
    fsl-first-data fsleyes incf-nidash-oneclick-clients mclaren-rhesus-macaque-atlas \
    mni-colin27-minc mni-colin27-nifti mni-icbm152-nlin-2009a mni-icbm152-nlin-2009b mni-icbm152-nlin-2009c \
    mriderfacer mrtrix3 mrtrix3-doc neurosynth-dataset odin openwalnut-qt4 rorden-mri-tutorial \
    sri24-atlas stabilitycalc stimfit utopia-documents vowpal-wabbit octave-psychtoolbox-3-nonfree \
    octave octave-symbolic octave-statistics octave-parallel octave-image octave-general \
    octave-financial octave-doc octave-dataframe octave-data-smoothing octave-control octave-communications \
    octave-biosig octave-bart virtual-mri-nonfree
do
echo "***** ${MYDPKG} *****"
su - save --command="ssh $DESTINATION -l root -t 'echo ; echo ${MYDPKG} ; date ; apt-get -y autoremove ; apt-get -y install ${MYDPKG}""
done
else
echo "No answer in ICMP ?"
fi
done

```

# Questions ?