ASL – Fall 2016

Gitlab, SW setup + Bash
Gitlab account and permissions
Create a repository

https://gitlab.inf.ethz.ch
Create a repository
Create a repository

Project name must be “asl-fall16-project”
Create a repository

The repository for this project is empty

If you already have files you can push them using command line instructions below.

Otherwise you can start with adding a README, a LICENSE, or a .gitignore to this project.

Command line instructions

Git global setup

```
git config --global user.name "Zsolt Istvan"
```

```
git config --global user.email "zsolt.istvan@inf.ethz.ch"
```

Create a new repository

```
git clone https://gitlab.inf.ethz.ch/zistvan/asl-fall16-project.git
```
Create a repository

Add all of us to the project as “Reporter”, so that we can check out the code.

- kkara
- mewaida
- zistvan
- clauderb
- dasidler
Once the repository has been created...

- Send an email to your TA (find him on the website) with subject “ASL Azure Voucher”
- First line: your short nethz login
- Second line: your microsoft email address on which you want the voucher

After you sent this email, you will get an automated response on the MS account; follow instructions there. Since we will enter your details by hand, this might take 1-2 days.
Installing memcached and memaslap
echo "Install build tools and memcached"

sudo apt-get install build-essential libevent-dev memcached

echo "Get and build memaslap (will take a while)"

#use exactly this version of libmemcached!

wget https://launchpad.net/libmemcached/1.0/1.0.18/+download/libmemcached-1.0.18.tar.gz

tar xvf libmemcached-1.0.18.tar.gz

cd libmemcached-1.0.18

export LDFLAGS=-lpthread

./configure --enable-memaslap && make clients/memaslap
Remarks on memcached and memaslap

• Installing memcached should be fast. Once installed, start it with:

  memcached -p 11212 -t 1

  – Run it with 1 thread (this is the default setup for all experiments)

• Building memaslap will take minutes, so we recommend building it once, and then scp'ing it to new machines.

  – To start the clients use (in clients/):

    ./memaslap -s <server>:11212 -T 64 -c 64 -o1 -S 1s -t <time> -F <workloadconfig>

    – We will provide you with example configuration files (define read-write mix and value size), and you can create your own quite easily.

    – Pipe output to a text file so that you can plot throughput, response times later

    – For more info: http://docs.libmemcached.org/bin/memaslap.html
Bash Tutorial
Comments from previous years

- “I had to stay up all night to run experiments!”
- “I cannot work on this from home…”
- “It took us more than 40hr./week to work on this milestone.”
- Solution is simple

Automate the experiments!
Bash Basics

- Bash is the command line interface used by most Linux systems
- On Windows: Use Putty to connect to a remote machine
- Most Bash-commands are simple programs that are executed
- Recommended for automating experiments
  - Feel free to use other scripting languages (Python, ...)

14
Running Bash Commands

command arg1 arg2

    # starts a command prompt returns as soon as the command finished

command arg1 arg2 &

    # runs the command in the background

command arg1 arg2 > out

    # runs the command, write standard out to file called "out"

command arg1 arg2 | command2 arg

    # runs the command and "pipes" the output into command2
MYVAR=foo
# set variable MYVAR to foo
export MYVAR=bar
# set MYVAR, child processes will see it as well
if [ $MYVAR = foo ]
then
    echo "MYVAR is foo"
elif [ $MYVAR = bar]
then
    echo "Bar"
else
    echo "Somethingisodd"
fi
Loops in Bash

for i in $(ls)
do
  echo item:$i
done

for i in `seq 1 10`
do
  echo $i
done

COUNTER=0
while [ $COUNTER -lt 10]
do
  echo The counter is $COUNTER
  let COUNTER=COUNTER+1
done
Running across multiple machines

• Use SSH and SCP
  - ssh darkoma@optimus.ethz.ch
    • Login to machine called optimus.ethz.ch as user darkoma
  - ssh darkoma@optimus.ethz.ch "ls"
    • Execute command "ls" on optimus as user darkoma
    • This is great for scripting!!!
  - scp project.jar darkoma@optimus.ethz.ch:~/asl
    • Copy project.jar to ~/asl on optimus

• Hints:
  - To use ssh and scp in scripts use public key for passwordless authentication
  - Use ssh -i <some-key> to use a different key-pair from your default one
Enabling Passwordless Login

• To enable passwordless login on local machine:
  
cat ~/.ssh/id_rsa.pub >>
~/.ssh/authorized_keys

• To enable passwordless login on remote machine:
  
  ssh-copy-id <username>@<machine>

• If there is no id_rsa.pub:
  
  ssh-keygen

• And don't give a password
ASL – Fall 2016

Gitlab, SW setup + Bash
Gitlab account and permissions
Create a repository

https://gitlab.inf.ethz.ch
Create a repository
Create a repository

Project name must be "asl-fall16-project"
Create a repository

The repository for this project is empty
If you already have files you can push them using command line instructions below.
Otherwise you can start with adding a README, LICENSE, or a .gitignore to the project.

**Command line instructions**

**git global setup**

```
git config --global user.name "Croat Server"
git config --global user.email "croat.croatlab@edu.etrs.ch"
```

**Create a new repository**

```
git clone https://gitlab.ethz.ch/栉柴fall16-project.git
```

---

6
Create a repository

Add all of us to the project as "Reporter", so that we can check out the code.

kkara
mewaida
zistvan
claudeb
dasidler
Once the repository has been created...

• Send an email to your TA (find him on the website) with subject “ASL Azure Voucher”
• First line: your short nethz login
• Second line: your microsoft email address on which you want the voucher

After you sent this email, you will get an automated response on the MS account; follow instructions there. Since we will enter your details by hand, this might take 1-2 days.
Installing memcached and memaslap
echo "Install build tools and memcached"

sudo apt-get install build-essential libevent-dev memcached

echo "Get and build memaslap (will take a while)"

#use exactly this version of libmemcached!

wget https://launchpad.net/libmemcached/1.0/1.0.18/+download/libmemcached-1.0.18.tar.gz

tar xvf libmemcached-1.0.18.tar.gz

cd libmemcached-1.0.18

export LDFLAGS=-lpthread

./configure --enable-memaslap && make clients/memaslap
Remarks on memcached and memaslap

• Installing memcached should be fast. Once installed, start it with:
  
  ```
  memcached -p 11212 -t 1
  ```

  - Run it with 1 thread (this is the default setup for all experiments)

• Building memaslap will take minutes, so we recommend building it once, and then scp'ing it to new machines.

  - To start the clients use (in clients/):

  ```
  ./memaslap -s <server>:11212 -T 64 -c 64 -o1 -S 1s -t <time> -F <workloadconfig>
  ```

  - We will provide you with example configuration files (define read-write mix and value size), and you can create your own quite easily.

  - Pipe output to a text file so that you can plot throughput, response times later

  - For more info: http://docs.libmemcached.org/bin/memaslap.html
Bash Tutorial
Comments from previous years

- “I had to stay up all night to run experiments!”
- “I cannot work on this from home…”
- “It took us more than 40hr./week to work on this milestone.”

Solution is simple

**Automate the experiments!**
Bash Basics

- Bash is the command line interface used by most Linux systems
- On Windows: Use Putty to connect to a remote machine
- Most Bash-commands are simple programs that are executed
- Recommended for automating experiments
  - Feel free to use other scripting languages (Python, ...)

14
Running Bash Commands

```
command arg1 arg2
    # starts a command prompt returns as soon as the command finished
command arg1 arg2 &
    # runs the command in the background
command arg1 arg2 > out
    # runs the command, write standard out to file called "out"
command arg1 arg2 | command2 arg
    # runs the command and "pipes" the output into command2
```
MYVAR=foo
# set variable MYVAR to foo
export MYVAR=bar
# set MYVAR, child processes will see it as well
if [ $MYVAR = foo ]
then
  echo "MYVAR is foo"
elif [ $MYVAR = bar]
then
  echo "Bar"
else
  echo "Somethingisodd"
fi
Loops in Bash

for i in $(ls)
  do
    echo item:$i
done

for i in `seq 1 10`
  do
    echo $i
done

COUNTER=0
while [ $COUNTER -lt 10]
  do
    echo The counter is $COUNTER
    let COUNTER=COUNTER+1
done
Running across multiple machines

- Use SSH and SCP
  - `ssh darkoma@optimus.ethz.ch`
  - Login to machine called optimus.ethz.ch as user darkoma
  - `ssh darkoma@optimus.ethz.ch "ls"`
  - Execute command "ls" on optimus as user darkoma
  - This is great for scripting!!!
  - `scp project.jar darkoma@optimus.ethz.ch:~/asl`
  - Copy project.jar to ~/asl on optimus

- Hints:
  - To use ssh and scp in scripts use public key for passwordless authentication
  - Use ssh -i <some-key> to use a different key-pair from your default one
Enabling Passwordless Login

- To enable passwordless login on local machine:
  cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

- To enable passwordless login on remote machine:
  ssh-copy-id <username>@<machine>

- If there is no id_rsa.pub:
  ssh-keygen

- And don’t give a password