Comp 290-088
Instructions for Using Surge
(Adapted from Don Smith's COMP 243 documentation)

Note: These instructions are written for the current network configuration that includes two "private" networks. See the link from the course web page for a diagram of the networks.

One machine on each network is designated to run the Apache web server (*daffy10* or *bugs30*) and one is designated to run SURGE (*brain10* or *speedy30*). Be sure you are using three machines on the ***same*** network for your experiments.

About Surge

The SURGE traffic generator can be run through a simple front-end shell script. SURGE's front-end is /usr/local/surge/surge.bash. To start SURGE, change directory to /usr/local/surge/ because you have write permissions there so you can retain the Surge.log file that records its activity during a run. (Also to avoid any problems with permission when you try to overwrite Surge.log, delete any older version of Surge.log before you start surge) Note that you only need to run one instance of SURGE.
IMPORTANT: Only one instance of SURGE can be running on a machine at a time. Be sure to check before attempting to start SURGE (%ps –ax | grep surge).

The SURGE shell script takes four positional parameters:
<destination machine name> <port> < number of concurrent threads> <elapsed time in seconds>
where:

1) **<destination machine name> <port>** give addressing information for SURGE to send requests.
2) **< number of concurrent threads>** controls the rate at which SURGE generates requests.
3) **<elapsed time>** the maximum time SURGE is to be run in seconds

Sample invocation:

```
/usr/local/surge/surge.bash daffy10 8080 10 7200
```

Testbed Configuration

A scheduling program that allows you to reserve time on a private network is available from the course web-page. The network reservation is necessary to prevent any concurrent use of the network so you can run the experiment under controlled conditions. Some additional notes are:

- As you see on the diagram, there are two "private" subnets added. We shall refer to them as the 192.168.10 subnet and the 192.168.30 subnet.

- The two subnets are identical in all respects, including machine specs and machine performance.
- You can only access any of the machines on these subnets from *nee*.
- Each subnet has a “router” to the subnet (which also runs the apache server).
- Your home directories on all these machines should be same as on *nee* (expect for the linux servers of course), i.e. they are mounted on all the machines.
- The subnets are completely isolated from each other – you work on any one subnet at any given time.
- Please honor your reservation time and those of others, i.e., use a subnet only after you have reserved it for exclusive access. Of course, at any time, you may continue to use nee for your development work, keeping in mind that others are using these machines, and hence performance testing should only be done on the subnets.

At any time, if you have questions about the network setup, please feel free to email Sushant Rewaskar <rewaskar_at_cs.unc.edu>

Running Surge

1) Log on *nee* using your department passwords. You can log on to the “private network” only from *nee*
2) Log onto *brain10* or *speedy30* from *nee*.
3) Go to directory “/usr/local/surge”
4) Make sure there is no old version of Surge.log. if present delete it.
5) Run surge using “/usr/local/surge/surge.bash ServerName ServerPort Threads Time”. E.g. /usr/local/surge/surge.bash daffy10 8080 10 7200
6) Copy Surge.log in your home directory

Analyzing Surge.log

Each line in Surge.log has the following fields

1) <client ID>,
2) <process ID>,
3) <session ID>,
4) <starttime(sec)>,
5) <starttime (usec)>,
6) <URL>,
7) <file size>,
8) <endtime (sec)>,
9) <endtime (usec)>

Sample Surge.log file

```
100 0 0 1108509363 133502 daffy100/177.txt 455 1108509363 136727
100 0 1 1108509363 183664 daffy100/1948.txt 224468 1108509363 213020
```

```
100 0 2 1108509363 233859 daffy100/3.txt 109 1108509363 236452
100 0 3 1108509363 283910 daffy100/243.txt 604 1108509363 286371
100 0 4 1108509363 333991 daffy100/160.txt 410 1108509363 336664
100 0 5 1108509363 384140 daffy100/611.txt 1426 1108509363 386133
100 0 6 1108509363 464146 daffy100/375.txt 868 1108509363 466585
100 0 7 1108509363 664552 daffy100/171.txt 442 1108509363 666726
100 0 8 1108509363 714550 daffy100/375.txt 868 1108509363 717042
100 0 9 1108509363 764859 daffy100/375.txt 868 1108509363 767594
100 0 2 1108509364 375545 daffy100/1919.txt 138100 1108509364 391454
100 0 7 1108509364 673488 daffy100/375.txt 868 1108509364 675494
100 1 0 1108509364 43595 daffy100/435.txt 995 1108509364 46676
100 1 1 1108509364 94276 daffy100/209.txt 536 1108509364 97213
100 1 2 1108509364 144140 daffy100/367.txt 854 1108509364 146610
100 1 3 1108509364 194224 daffy100/326.txt 776 1108509364 196705
100 1 4 1108509364 244315 daffy100/985.txt 2753 1108509364 247272
100 1 5 1108509364 294703 daffy100/375.txt 868 1108509364 297118
100 1 6 1108509364 344794 daffy100/279.txt 678 1108509364 347521
100 1 7 1108509364 394935 daffy100/375.txt 868 1108509364 396998
100 1 8 1108509364 445610 daffy100/650.txt 1530 1108509364 447619
100 1 9 1108509364 495769 daffy100/375.txt 868 1108509364 497920
100 1 4 1108509365 285741 daffy100/776.txt 1987 1108509365 305817
100 1 4 1108509365 306065 daffy100/1568.txt 11518 1108509365 309609
```

So in line # 1:

```
100 0 0 1108509363 133502 daffy100/177.txt 455 1108509363 136727
```

1) <client ID> = 100
2) <process ID> = 0
3) <session ID> = 0
4) <starttime(sec)> = 1108509363
5) <starttime (usec)> = 133502
6) <URL> = daffy100/177.txt
7) <file size> = 455
8) <endtime (sec)> = 1108509363
9) <endtime (usec)> =136727