System Imager

SystemImager and BitTorrent: a p2p approach to large scale OS deployment

Andrea Righi <a.righi@cineca.it>
Bernard Li <bli@bcgsc.ca>
Brian Elliott Finley <brian@thefinleys.com>
Erich Focht <efocht@hpce.nec.com>

LinuxTag 2007 (May 29 – Jun 2)

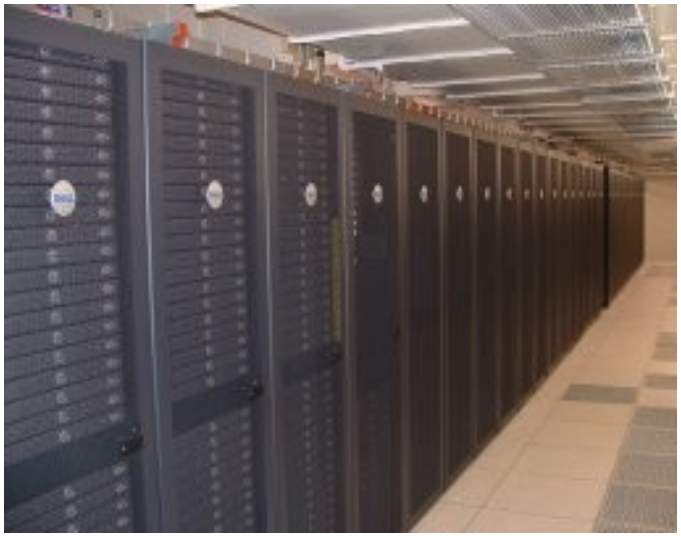
System Imager

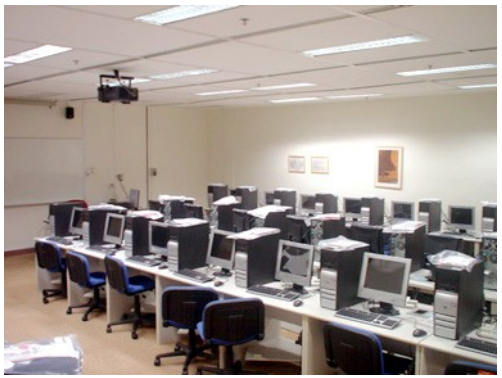
The problem

- You have to install a lot of computers
  - PC labs,
  - Server farms,
  - HPC clusters,
  - Complex grid-computing environments,
  - Etc.
- But you don't have enough time!

System Imager

A practical example





System Imager

The tool

- SystemImager is a software which automates GNU/Linux installs, software distributions and production deployment

System Imager

Major Goals

- Support all Linux distributions
- Support a large number of architectures
- Make it easy to add support for new distro and architectures
- Make it solve massive installation problems
- Create a centralized point of installation and maintenance

System Imager

What does it do for me?

- System Installation
- System Updates
- Build replicants of machines
- File system or block device migration

System Imager

Advantage of SystemImager

- File-oriented approach
  - Distribution agnostic
  - Hardware independence
  - Filesystem independence
  - Plain filesystem dump: exclude swap space or unused partitions
  - Block device independence
  - Live customization (manipulate cloned filesystems directly)

Basic concepts

System Imager

Basic concept: image

- Image:
  - Live snapshot of a machine containing files and directories from the root of that machine's filesystem
  - *chroot*-able filesystem stored in `/var/lib/systemimager/images/$NAME`
  - Examples:
    - `/var/lib/systemimager/images/RHEL4`
    - `/var/lib/systemimager/images/Debian_Etch`
    - `/var/lib/systemimager/images/HPC_1.0`
    - `...`

Basic concept: image server

- Image Server:
  - a server that has all the images available for the installation
  - "*Jukebox*" of images

Basic concept: client

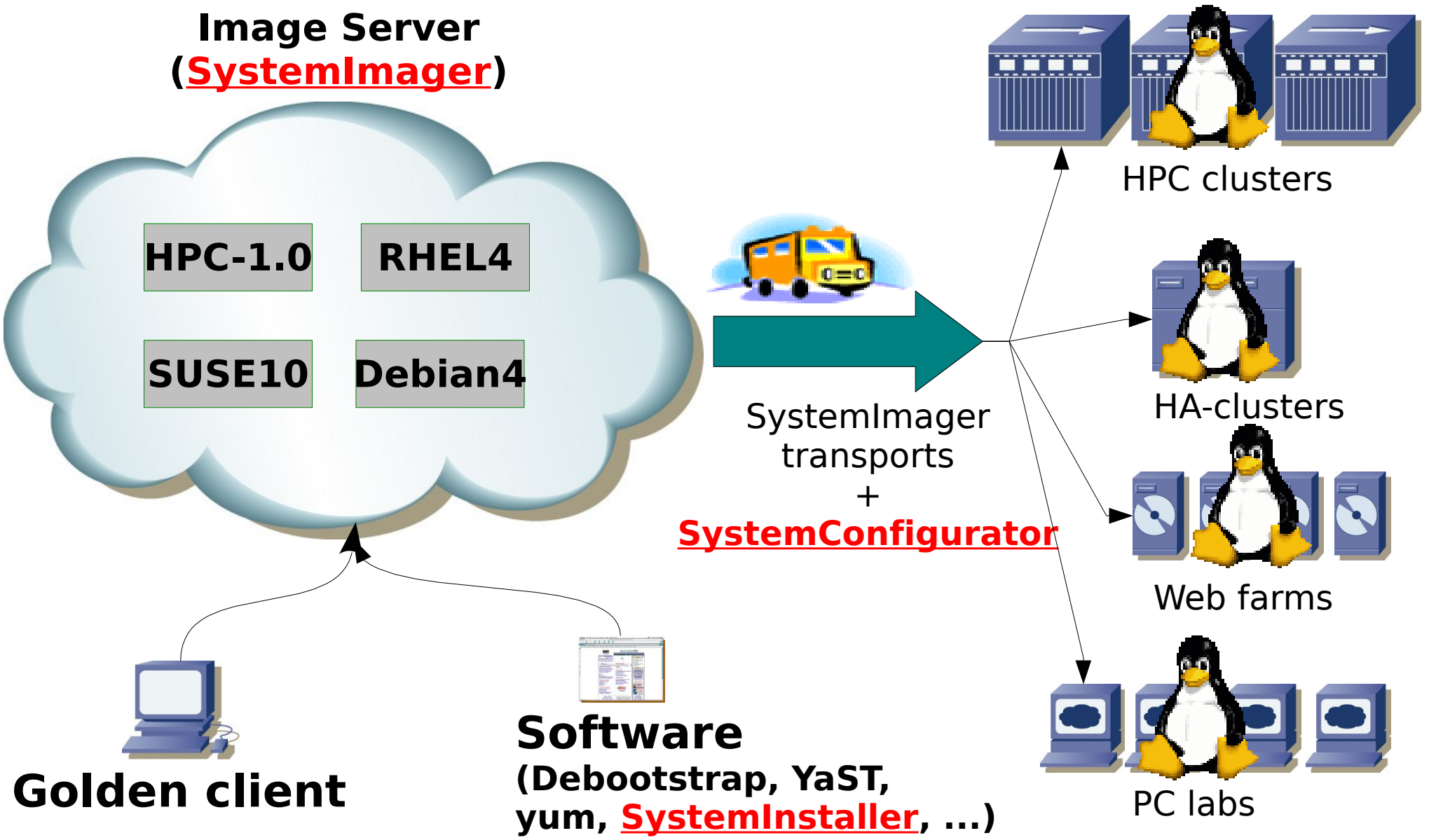
- Client:
  - a machine to be auto-installed with a (single) selected image
  - Example: the dancing penguins are the clients :-)

System Imager

Basic concept: transport

- Transport:
  - the protocol used to distribute images from the image server to the clients
  - *push/pull/p2p* approach
  - Examples:
    - rsync, multicast, SSL, BitTorrent, ...

Application stack (SIS)



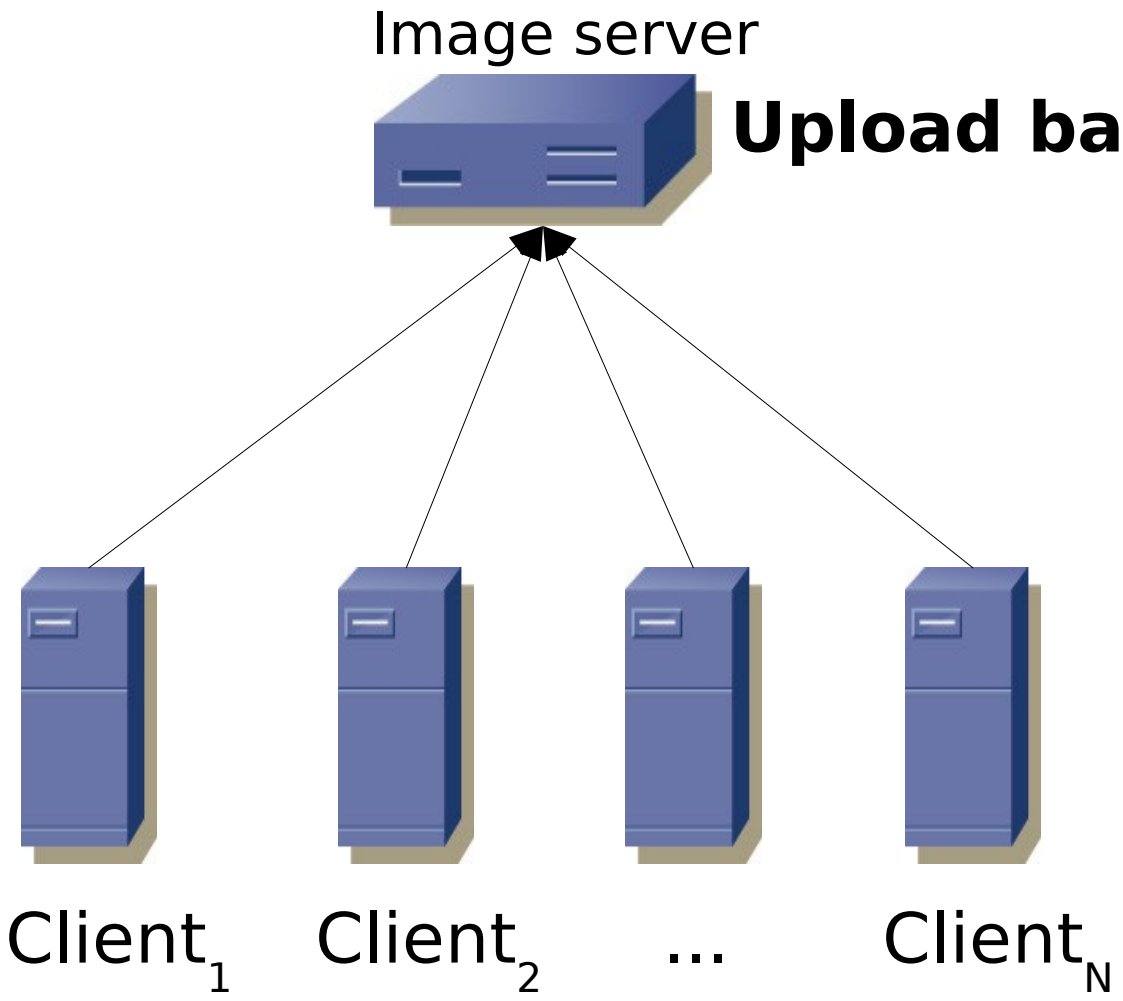
Transports

System Imager

rsync transport

- rsync (plain / SSL encrypted):
  - Client-server approach
    - Limited in scalability
    - Limited in reliability with a lot of clients
  - Max Theoretical Bandwidth: Us / N

rsync diagram



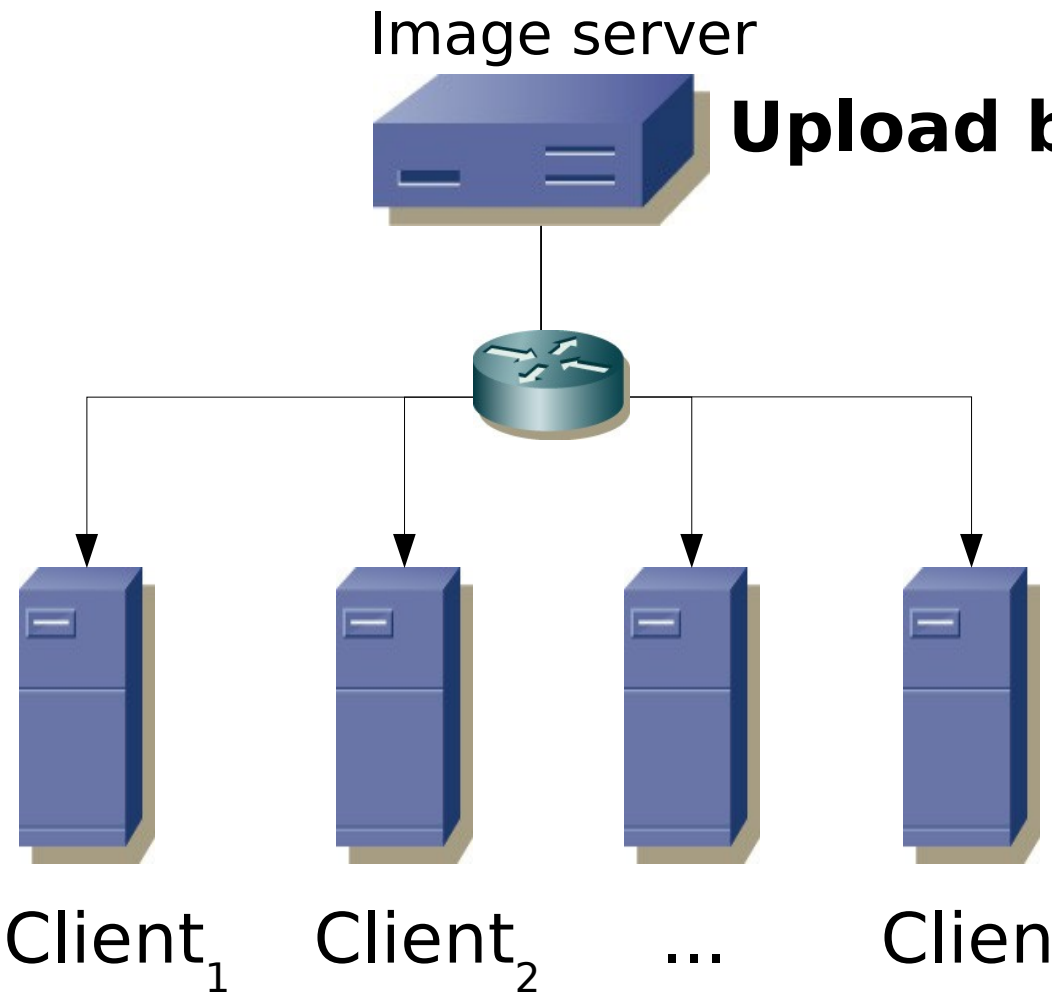
Download bandwidth = U_s / N

System Imager

Multicast (Flamethrower) transport

- Flamethrower:
  - Multicast approach:
    - Perfect scalability
    - But limited in reliability
  - Max Theoretical Bandwidth: *Us*

Multicast (Flamethrower) diagram



Download bandwidth = U_s

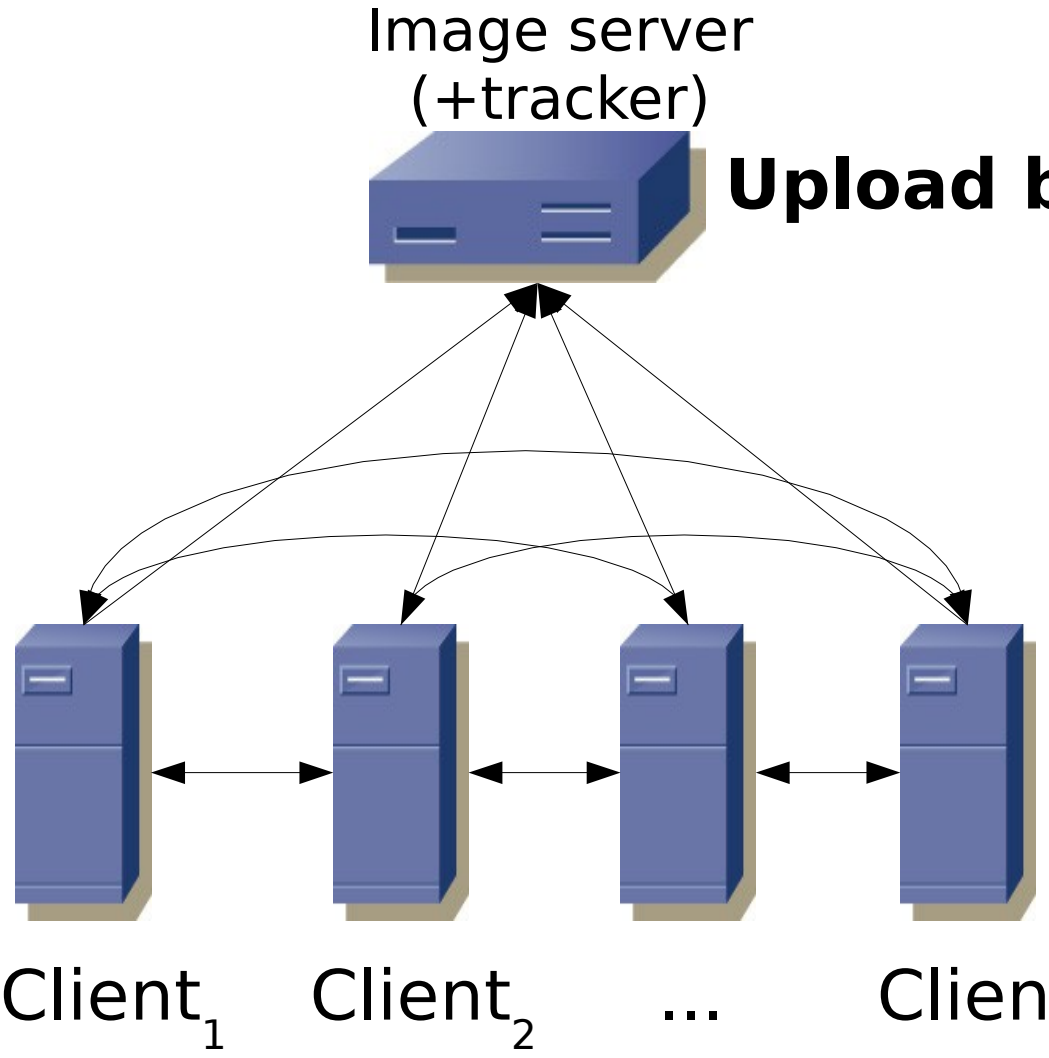
BitTorrent overview

- BitTorrent is a TCP/IP p2p oriented protocol designed for transferring files
- Peers connect to each other directly to send and receive chunks of data
- There is a central server (tracker) which coordinates the action of all such peers
- The tracker does not have any knowledge of the contents of the files being distributed
- Users upload (*transmit outbound*) at the same time they are downloading (*receiving inbound*)

BitTorrent transport

- BitTorrent:
  - p2p approach: scalability && reliability
  - Qiu and Srikant model
    - Total upload rate: $\mu(\eta x(t) + y(t))$
  - *Steady state:*
    - x(t) downloaders => $\frac{dx(t)}{dt} = 0$
    - y(t) uploaders => $\frac{dx(t)}{dt} = 0$
  - Max Theoretical Bandwidth:
    - *Us (steady state)*

BitTorrent diagram



Download bandwidth = U_s (in steady state)

BitTorrent fun!

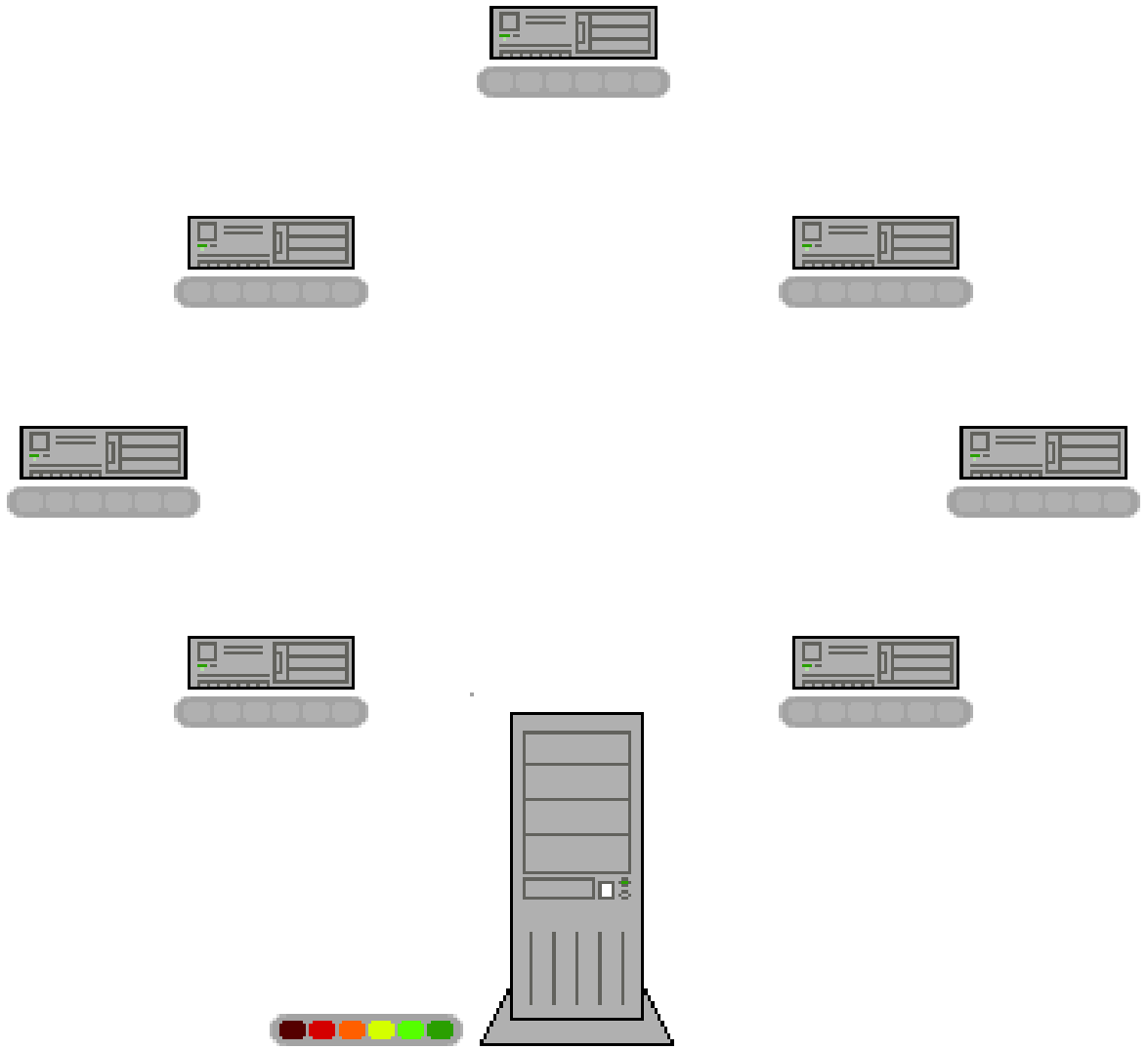


Image server

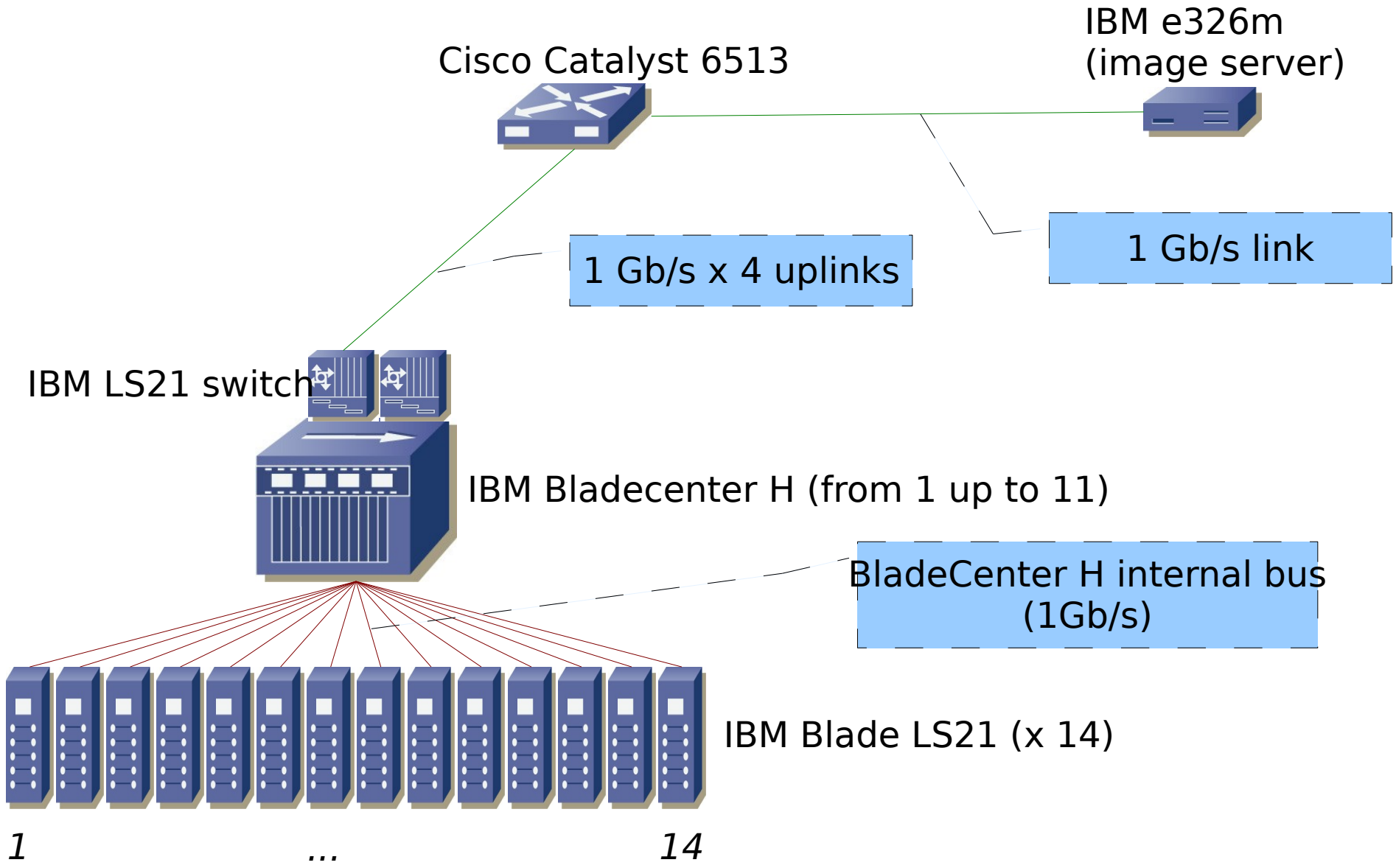
Testbed environment

BCX/5120 cluster @ CINECA

- IBM BCX/5120, with 5120 cores, is the largest computer in Italy for Scientific Computing
- 2 dual-core AMD Opteron(tm) 2.4GHz, 8GB RAM per node
- It is the 44th most powerful computer in the world (TOP500).

System Imager

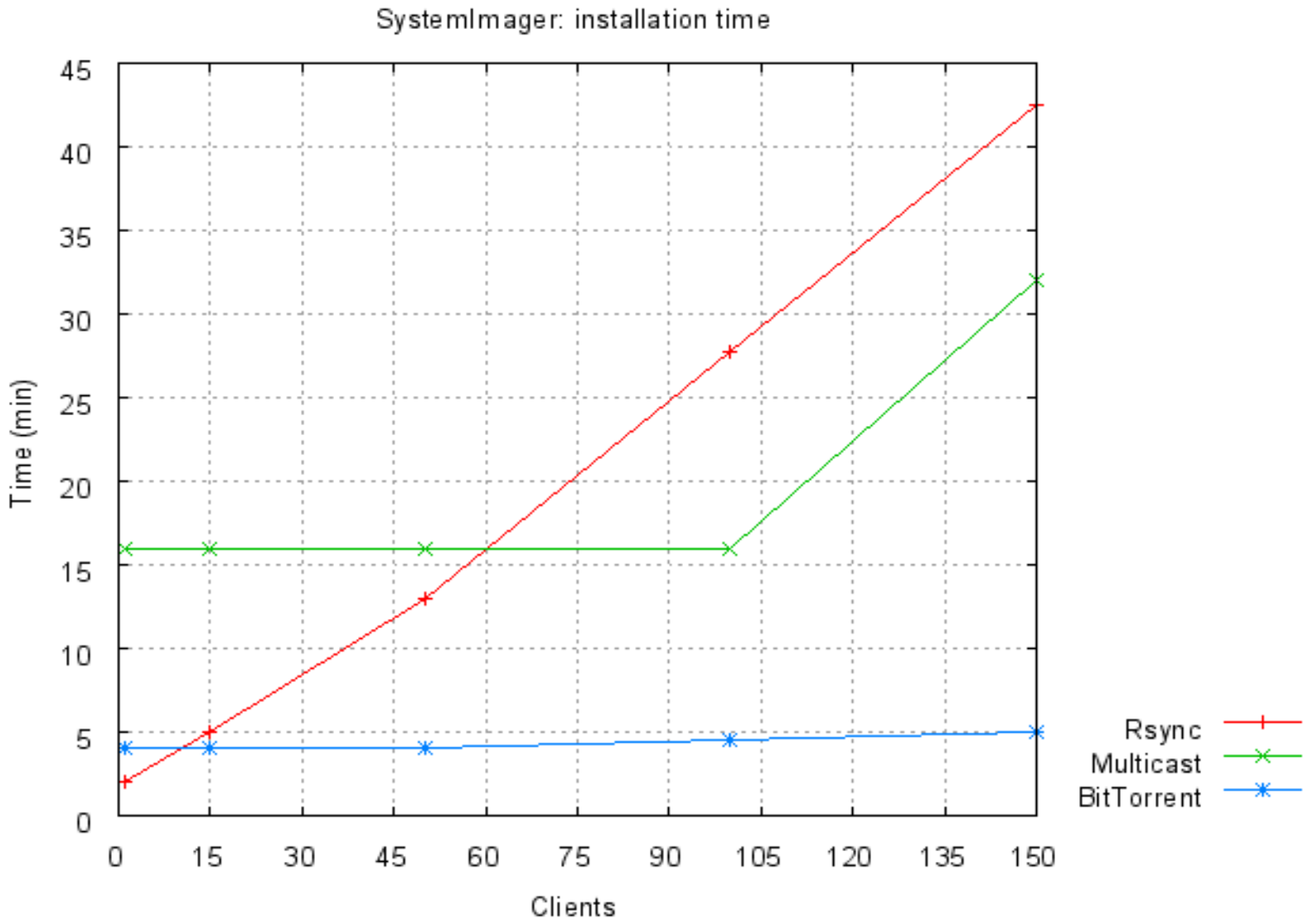
BCX network topology



Experimental results

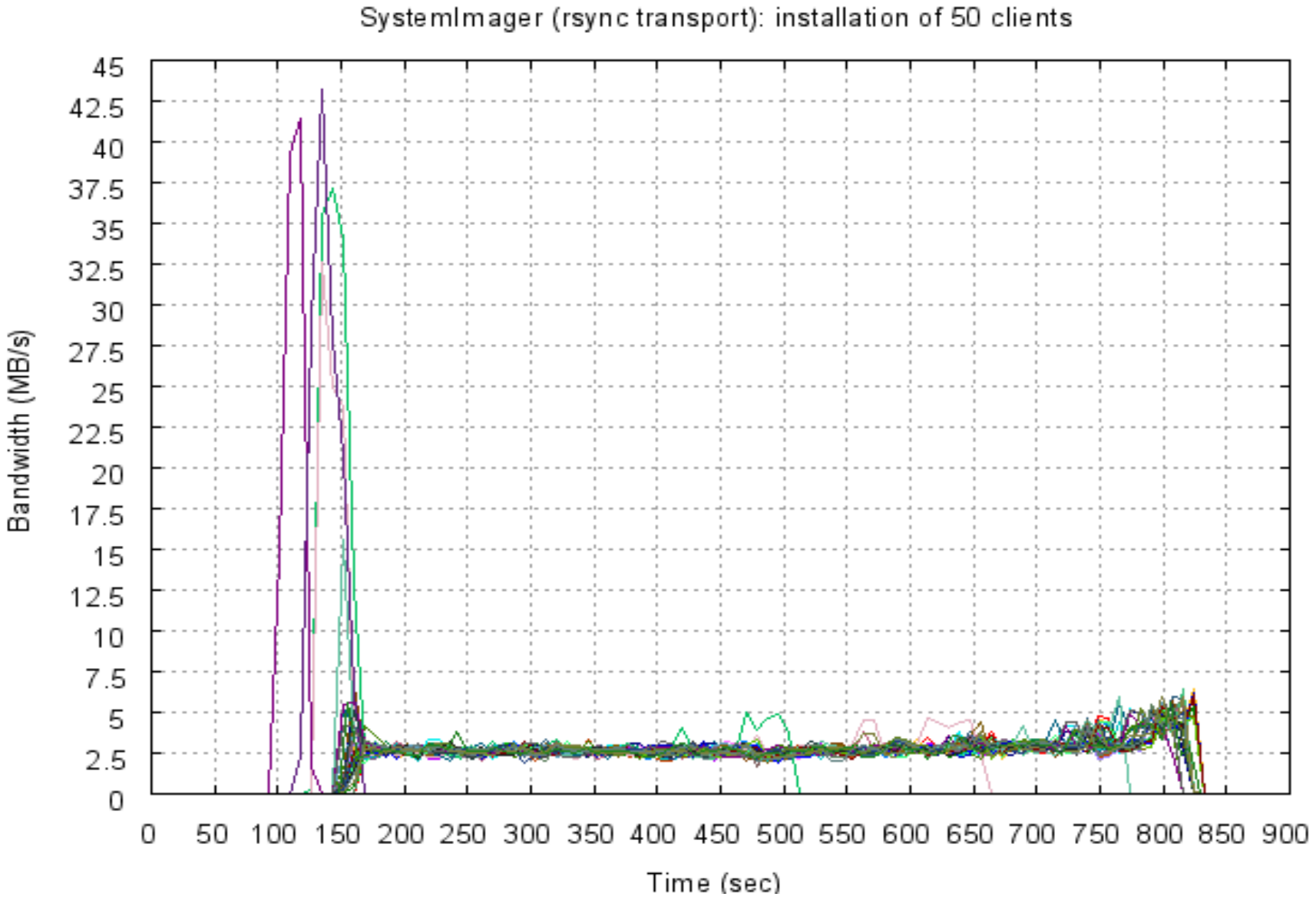
System Imager

Deployment time



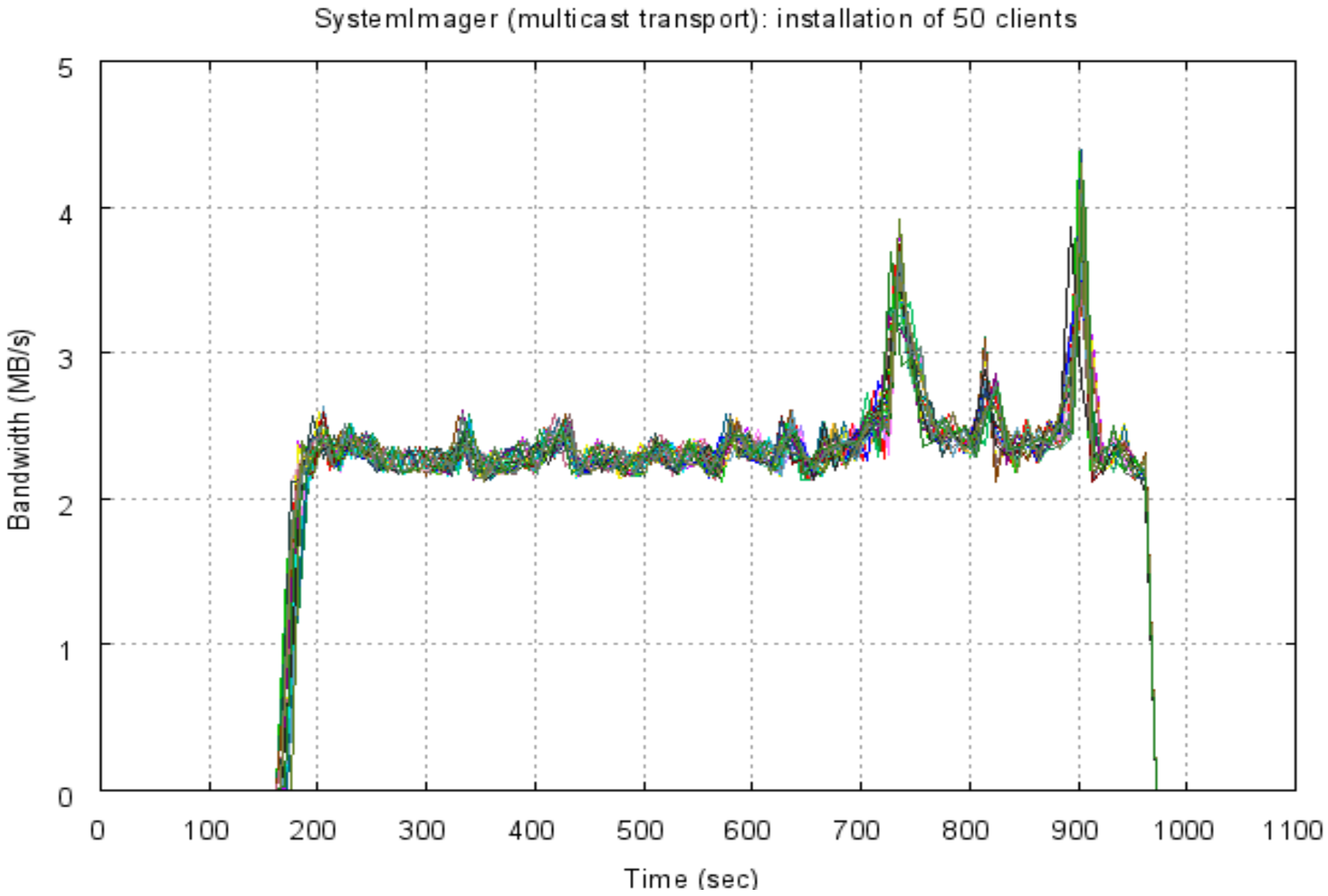
System Imager

Rsync: 50 clients (download rate)



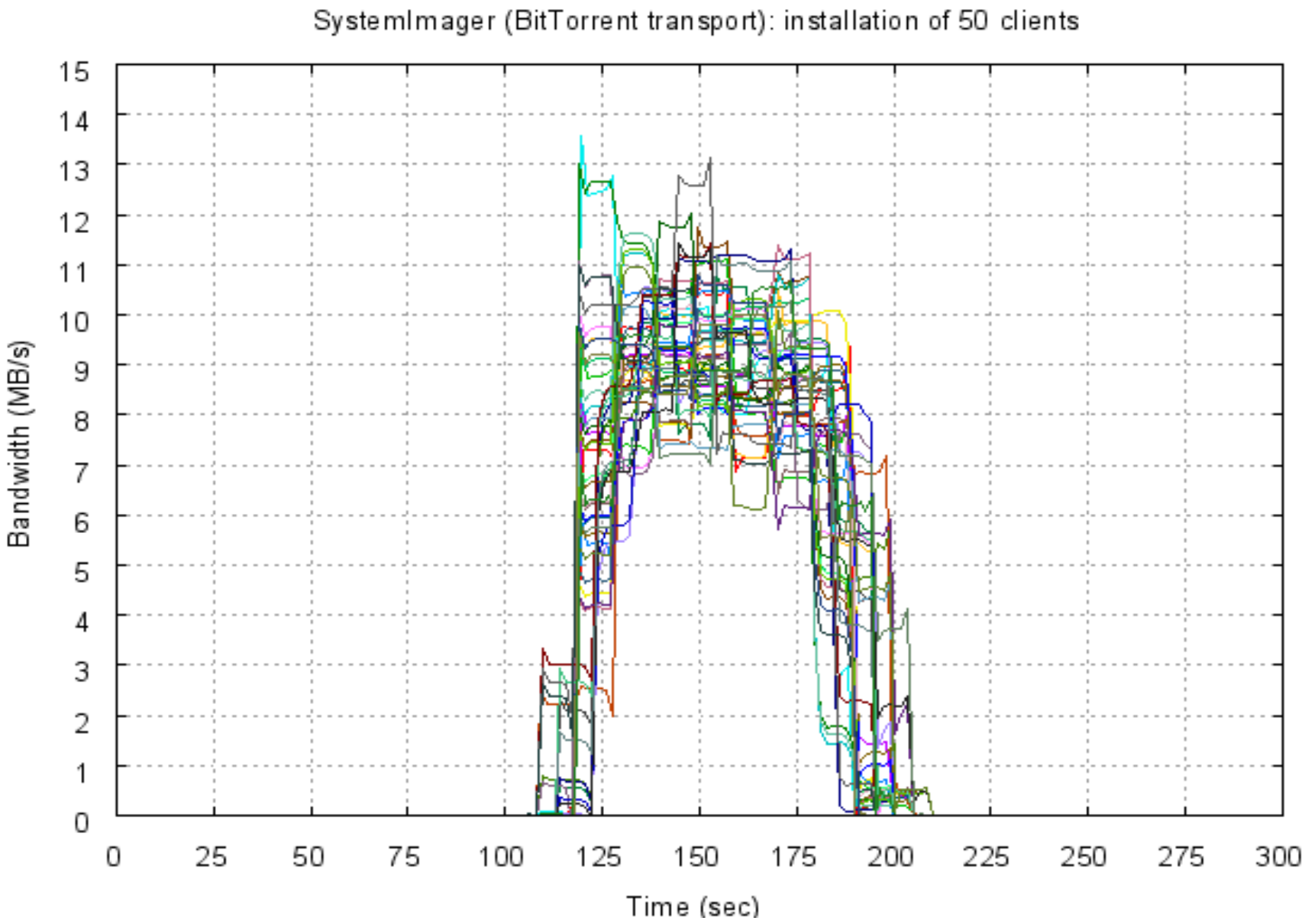
System Imager

Multicast: 50 clients (download rate)

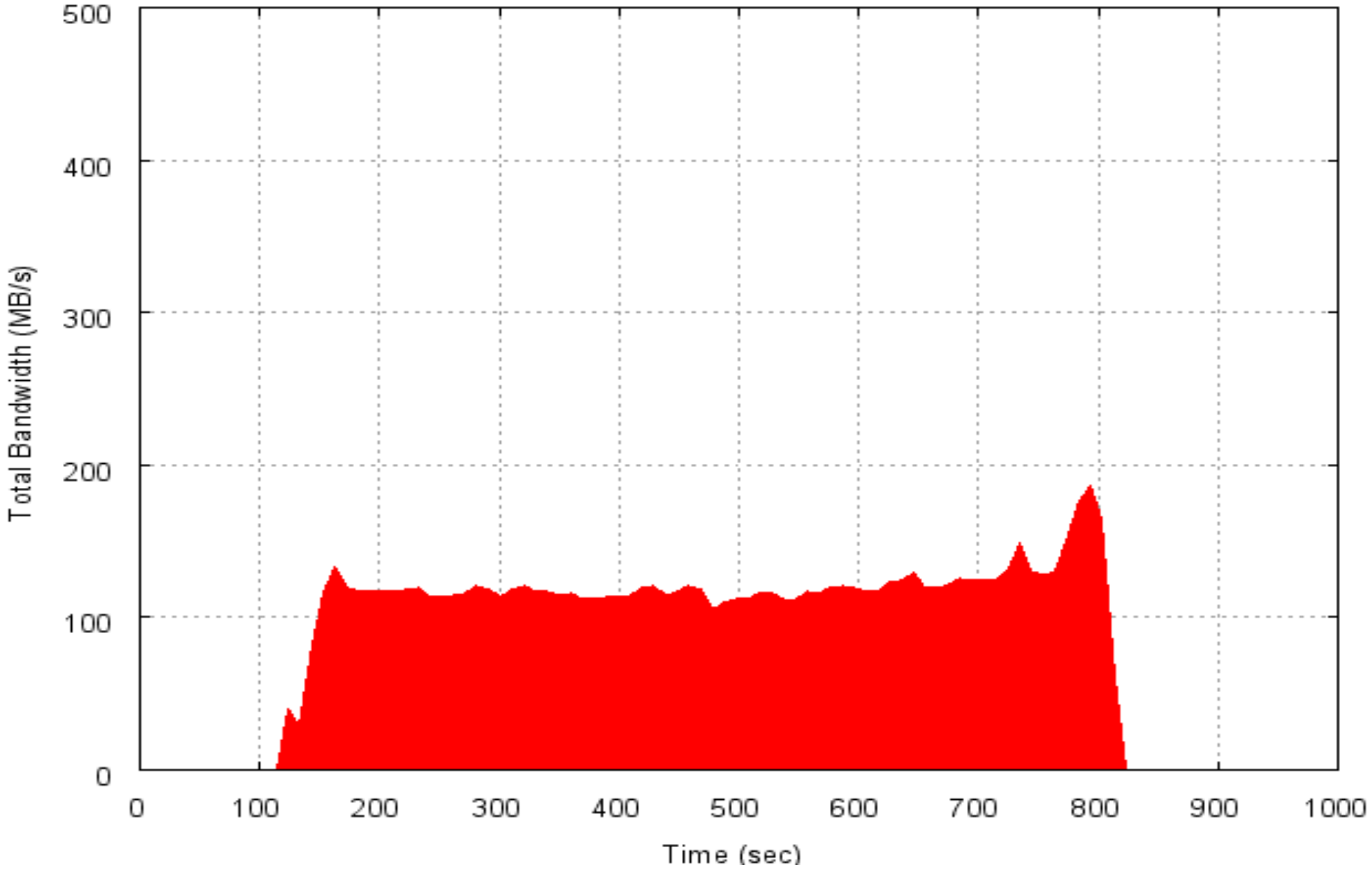


System Imager

BitTorrent: 50 clients (download rate)

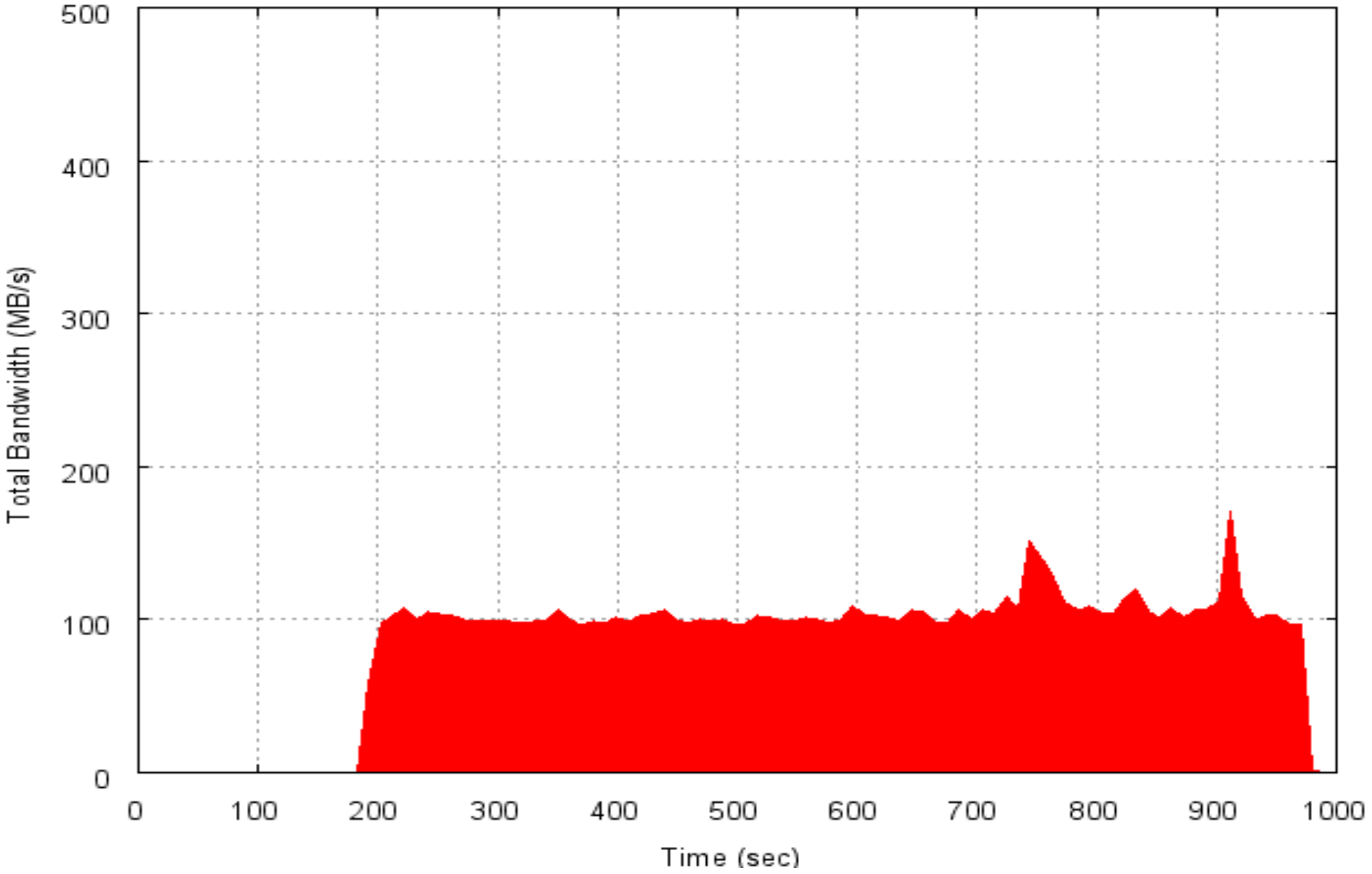


Rsync: 50 clients (aggregated bandwidth)

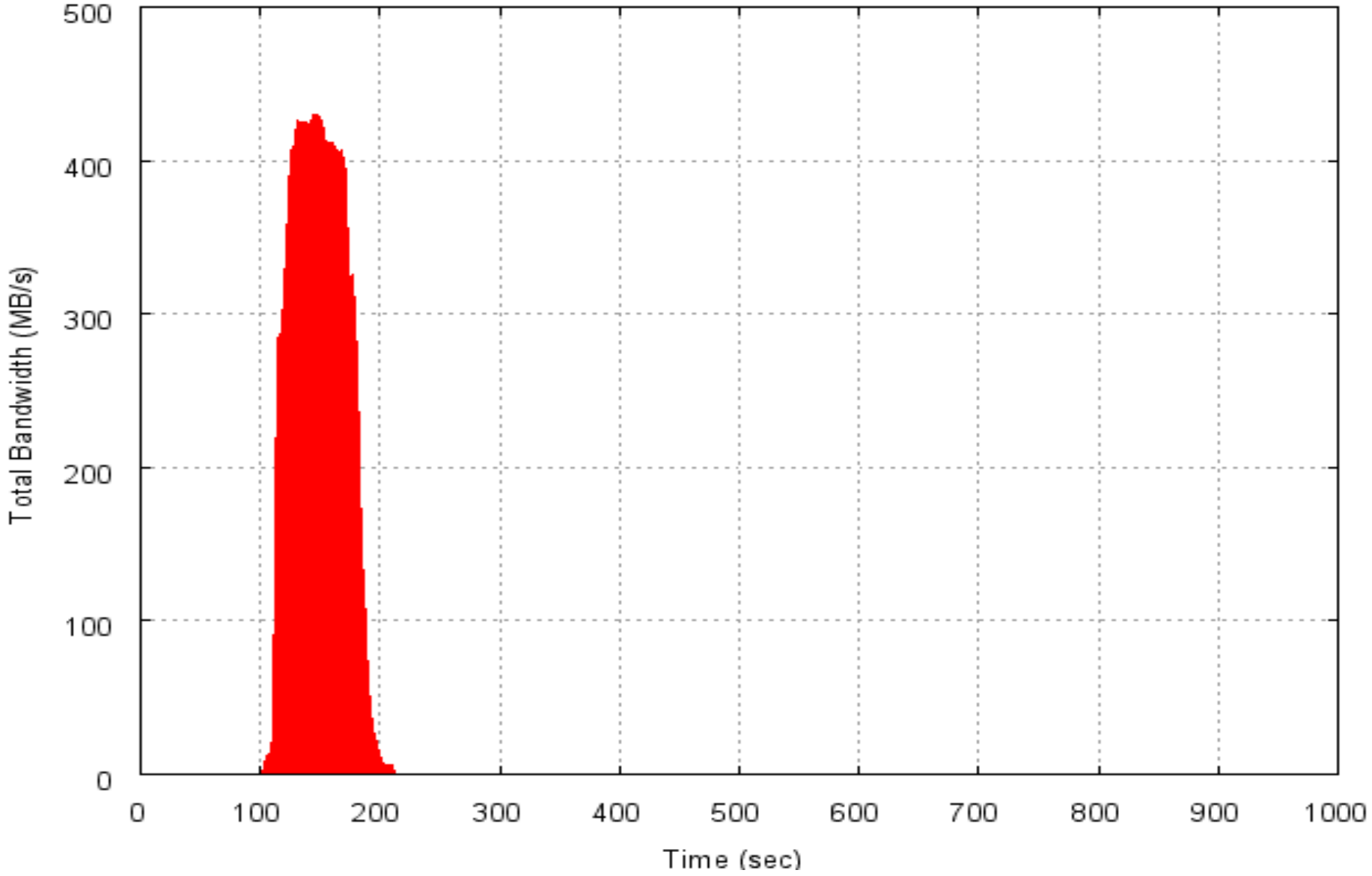


System Imager

Multicast: 50 clients (aggregated bandwidth)



BitTorrent: 50 clients (aggregated bandwidth)



Conclusion

Advantages

- Quicker deployment of images
- Safer deployment (better error handling)
- Less load on the image server
  - no need to buy a powerful machine

Disadvantages

- No time to have a coffee while the clients are imaging
- More disk space consumption!
  - tarballs of images
- Images and tarballs must be kept in-sync
  - Re-generate tarball and .torrent at each image change

Future work

- Optimize performance in LAN environments and dedicated HPC networks
- Improve security (encryption of BT tarballs)
- Virtual cluster deployment (re-imaging using the same physical resource pool)
- Exploit the p2p approach to create distributed and redundant repositories of custom image
- Use BT transport also for updates (pushing changes/differences of images) => a path to image version management

System Imager

References

- Web:
  - http://www.systemimager.org
- Mailing list:
  - sisuite-users@lists.sourceforge.net
  - sisuite-devel@lists.sourceforge.net
- IRC:
  - #sisuite (irc.freenode.net)

Tank you for attending!!!