Optimizing Boot speed

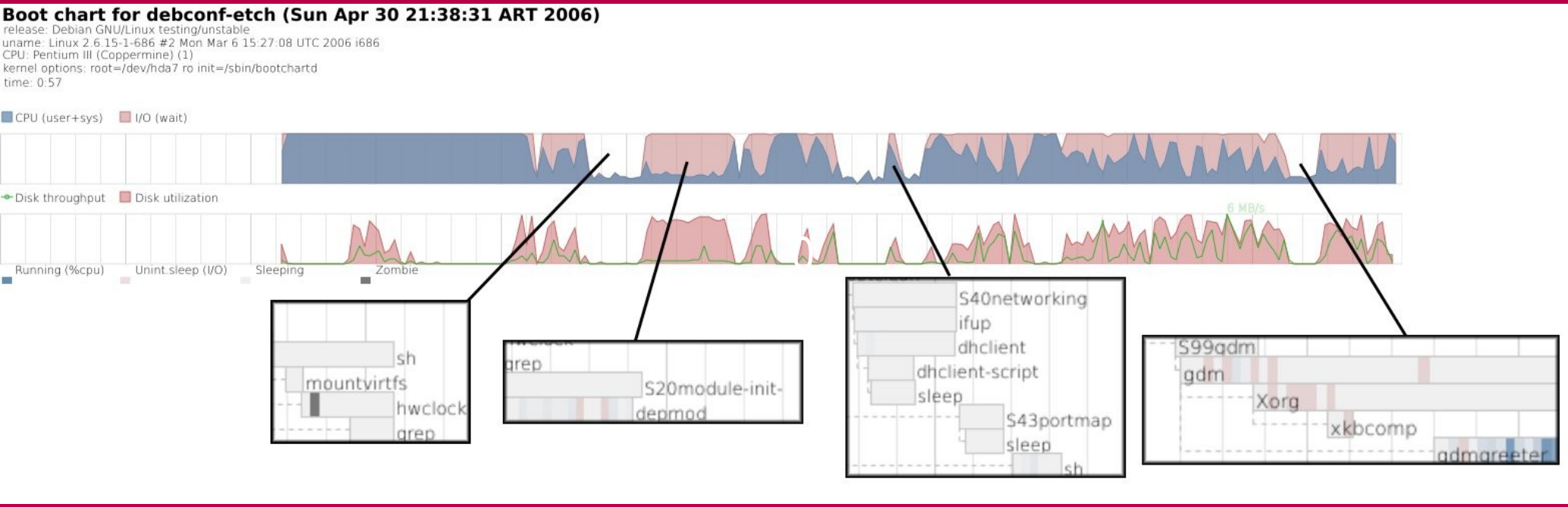
Some experiences and thoughts

Margarita Manterola
marga@debian.org

Times for different distributions

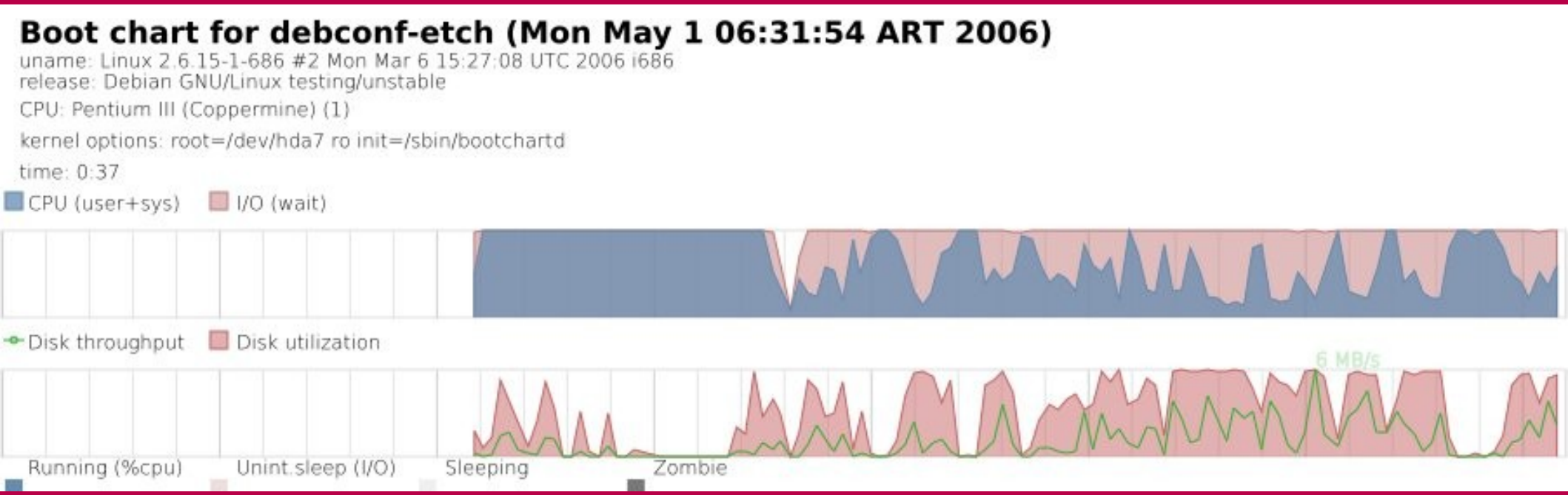
- Completely tweaked Etch: 0:37
- Configured Gentoo: 0:48
- Freshly installed Etch: 0:57
- Freshly installed Sarge: 0:59
- Freshly installed Dapper: 1:05
- Freshly installed Gentoo: 1:10
- Freshly installed Breezy: 1:33

What takes the time away



- Setting up the hardware clock
- Running depmod
- Starting the network
- Starting gdm

Changes and effects



- Setting up the network in the background: 2 seconds
- Removing depmod from the boot process: 2 seconds
- Running hwclock in background: 6 seconds
- Pointing /bin/sh to /bin/dash: 6 seconds
- Using parallel starting for services of the same priority while rearranging the scripts: 2 seconds
- Rearranging services so that the cpu is never idle: 2 seconds

Problems and possible solutions

- To parallelize, we need to know what has to be setup first: use script dependecies
- It's not a good idea to have to edit /etc/init.d/rc to change behaviour: have a conf file for boot process
- The boot process is too verbose: make scripts only output their success (or failure), unless otherwise configured