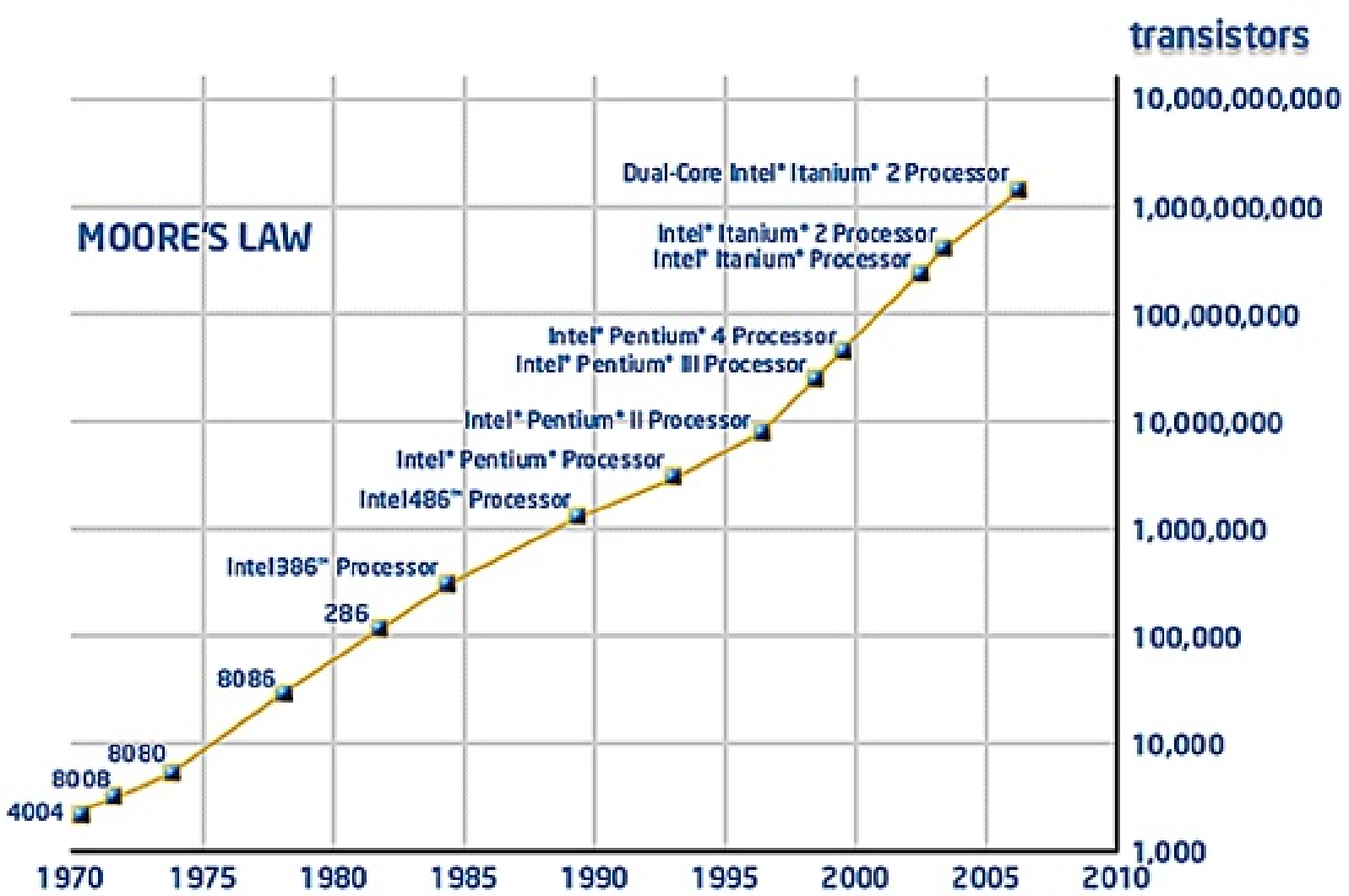
Cgroups as a Resource Management Technique

(C) 2012 Vladimir Zapolskiy <vz@mleia.com>

Functional	Mainframes	Servers	Workstations Tablets Smartphones
Embedded	Prototypes Research Conceptual Doomsday device	Medical Industrial Telecom Aerospace Military	Game Consoles SmartTV Set-top boxes Home Connectivity Automotive
	Exceptional $< 10^3$	Distributed	Massively Produced $> 10^7$

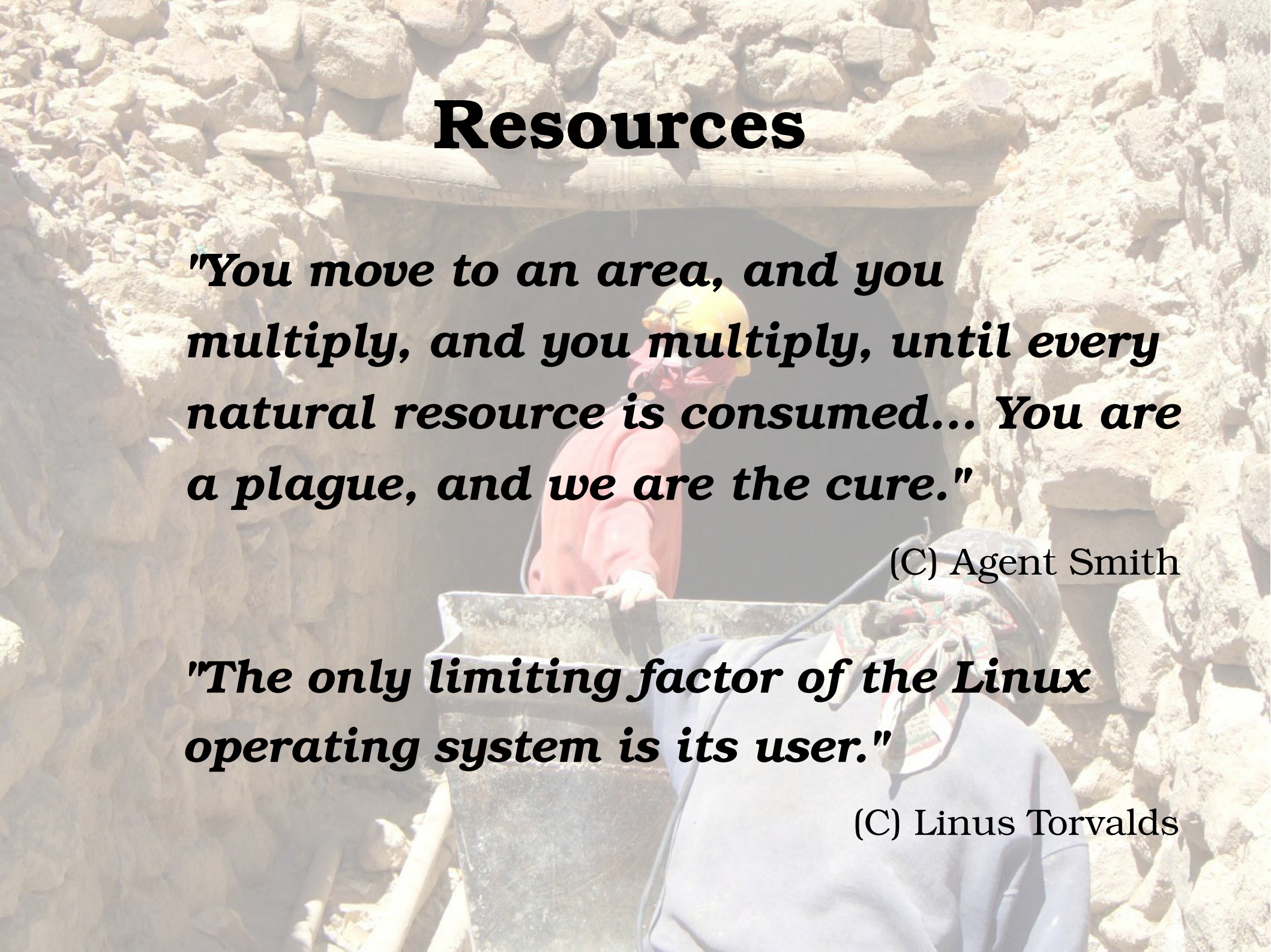
Massively produced devices have to be cheap, and what's going to be sacrificed then?



- **Network capacity**
- **Pixels per dollar**
- **CPU productivity**
- **Battery charge capacity**
- **Mass storage volume**
- **Disc I/O speed**
- **RAM access speed**
- **etc.**

Problems:

- **How to make a massively produced cheap and functional device today?**
- **How to maximize yield from potential in a given device?**
- **How to define acceptable balance of functionality, price and time-to-market for future devices?**



Resources

"You move to an area, and you multiply, and you multiply, until every natural resource is consumed... You are a plague, and we are the cure."

(C) Agent Smith

"The only limiting factor of the Linux operating system is its user."

(C) Linus Torvalds

Linux? Control Groups!

The first step to start learning about cgroups is to read ***linux/Documentation/cgroups.txt***

The main traits to emphasize:

- hierarchical tree-like cgroups layout
- modular extensive set of resource and isolation controllers: freezers, CPU sets, RAM memory, swap memory, CPU slices, real-time priority, block I/O, network I/O etc.
- cgroups can be set up and tasks can be operated immediately from shell

cgroups + proc connector = N900 success,
see */usr/share/policy/etc/current/syspart.conf*

Tortured knowledge:

- proc connector reports may be too late
- task nice level shouldn't be touched ever
- cgroup hierarchy layout is important
- task I/O ought to be limited by cgroups
- optimal shape of RAM/swap memory filling curve is unknown
- virtual memory overcommit allowance leads to asynchronous OOM happenings
- OOM is a mess even worse than expected

Opportunities and Unsolved Problems

- Cgroup partition change of a process under OOM circumstances
- Analysis and elimination of performance bottlenecks while moving processes between arbitrary cgroups
- Cgroup aliases?
- Power consumption per task controlled by cgroups
- Create a supported by community application, which utilizes proc connector to dynamic cgroups closing
- etc.


Ask questions

Thank you!

Vladimir Zapolskiy <vz@mleia.com>