



smem: understanding memory usage

Matt Mackall
Selenic Consulting
with support from CELF

The Accounting Problem

- We save memory by sharing it between processes
- ...but we count that memory multiple times when reporting it
- ...and we allocate more memory than is actually available
- The numbers don't add up!
- Users and developers can't get a good sense of how memory is used
- They end up bailing out the system by throwing more memory at it

Pagemap and friends

- In 2007, I attacked this problem from the kernel side with pagemap
- The pagemap interface exposes the mapping from virtual to physical memory and other details
- Along the way, two new concepts:
 - PSS (Proportional Set Size)
a mapping's fair share of shared memory
 - USS (Unique Set Size)
a mapping's non-overlapping memory usage
- ...and some proof-of-concept graphical tools

And then...

- I submitted pagemap support to the mainstream kernel
- Got lots of contributions from other developers
- Added direct PSS calculation
- Redesigned most of the code and interfaces
- Many embarrassing bugs added and discovered
- Linus: “This is crap!”
- Along the way, all my demo tools broke
- No one published tools for users!

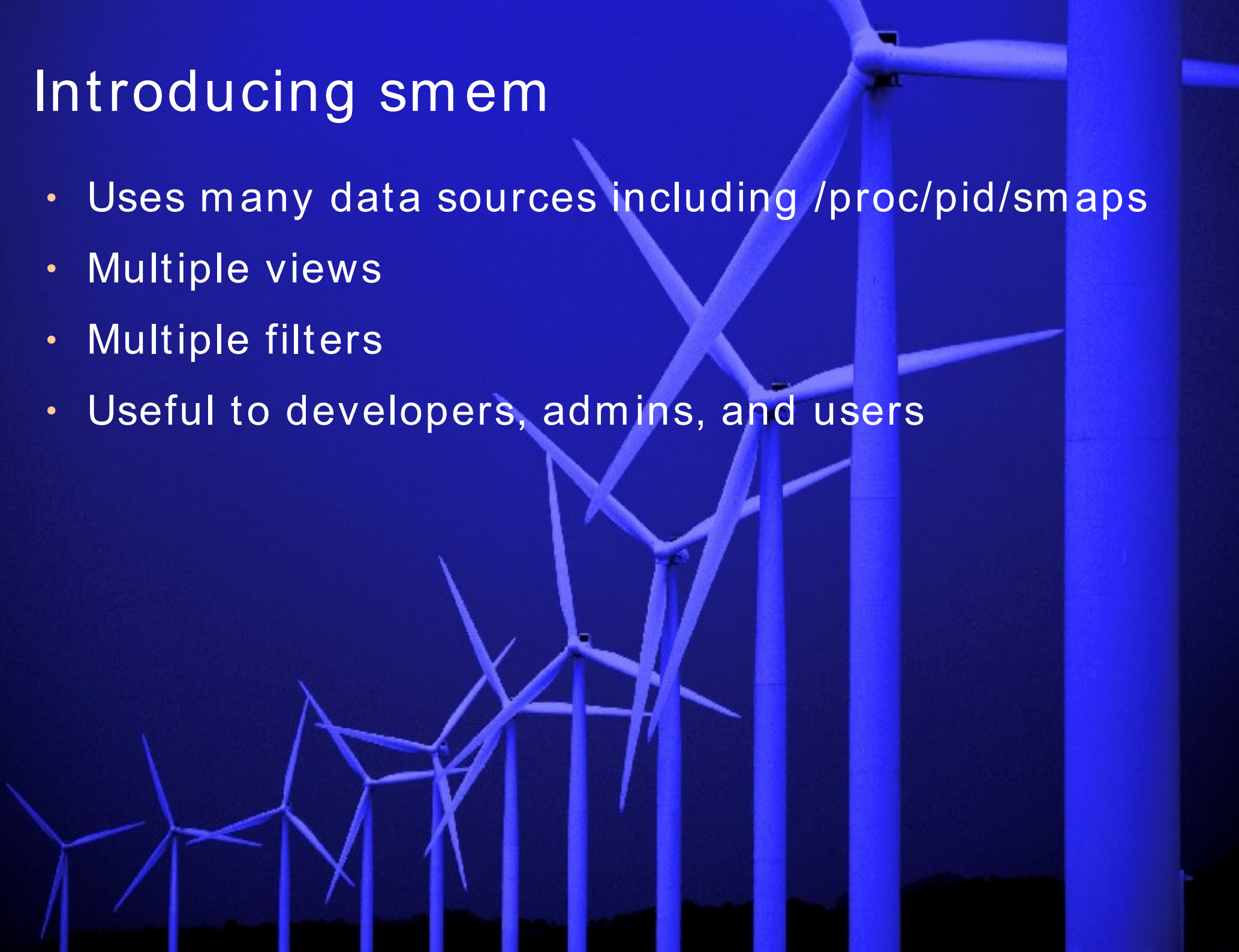
Take 2

- Let's write a useful tool and hope it catches on
- Let's submit a talk proposal to make sure I actually work on it
- Let's put visualization in the title to give ourselves a challenge



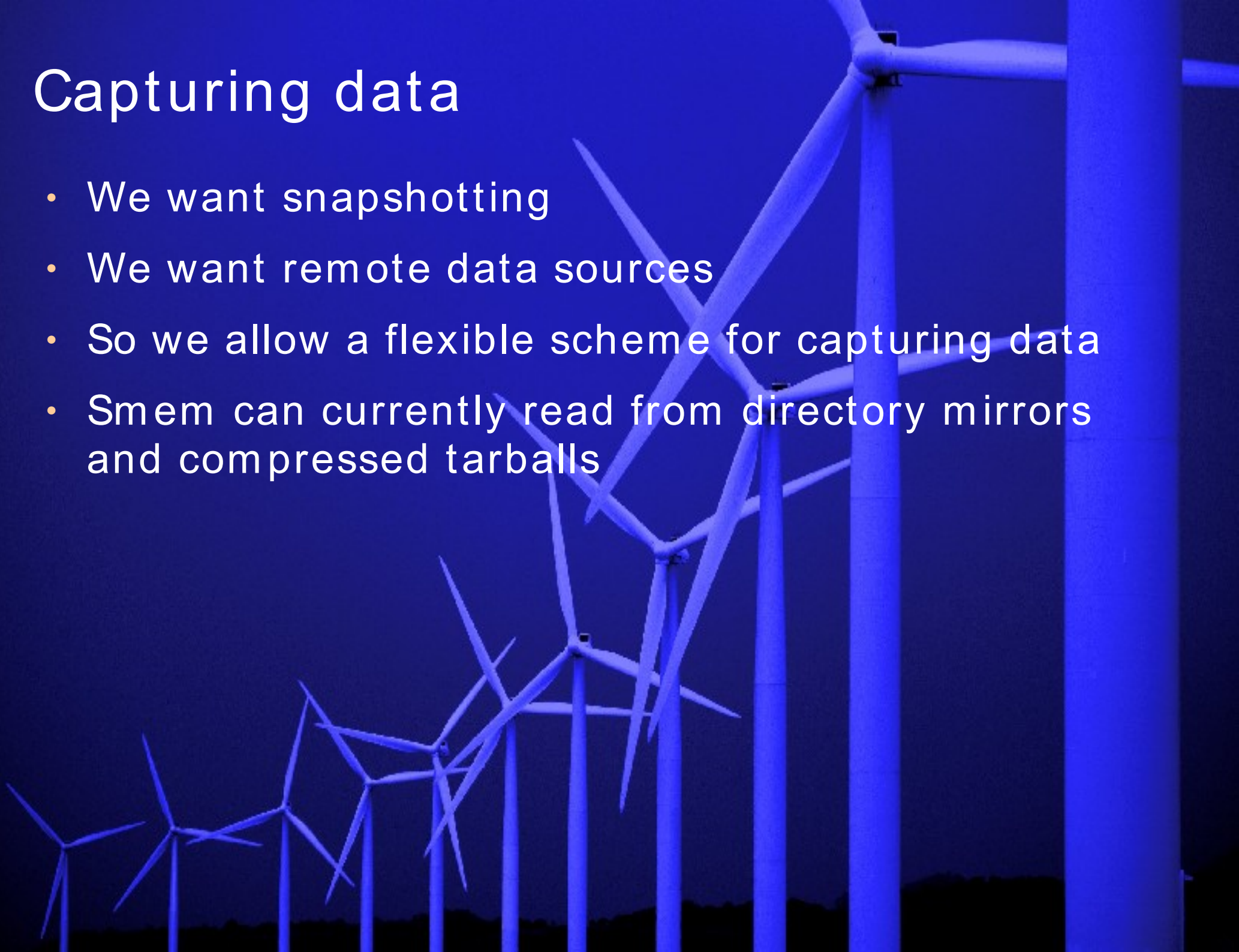
Introducing smem

- Uses many data sources including `/proc/pid/smaps`
- Multiple views
- Multiple filters
- Useful to developers, admins, and users



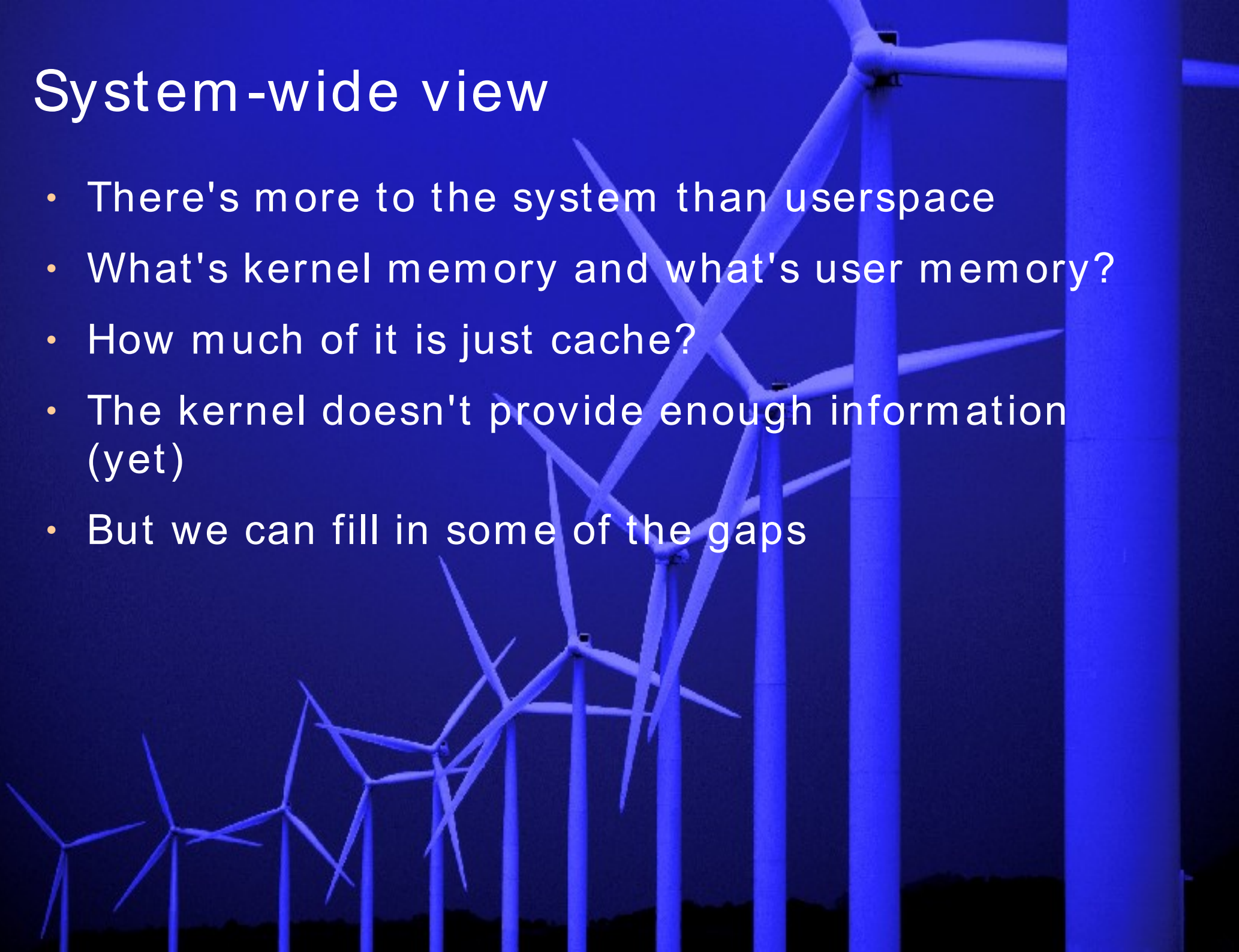
Capturing data

- We want snapshotting
- We want remote data sources
- So we allow a flexible scheme for capturing data
- Smem can currently read from directory mirrors and compressed tarballs



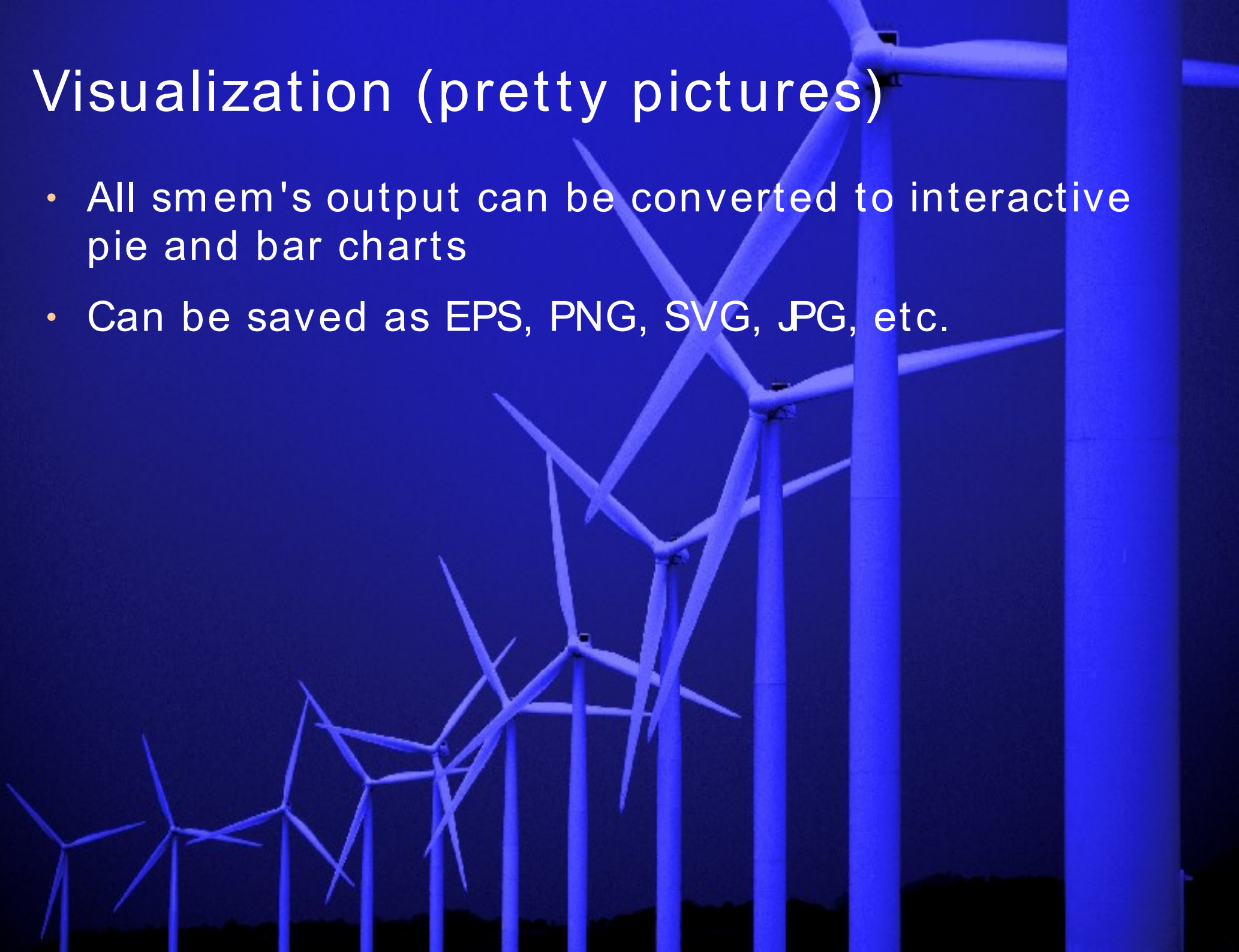
System-wide view

- There's more to the system than userspace
- What's kernel memory and what's user memory?
- How much of it is just cache?
- The kernel doesn't provide enough information (yet)
- But we can fill in some of the gaps



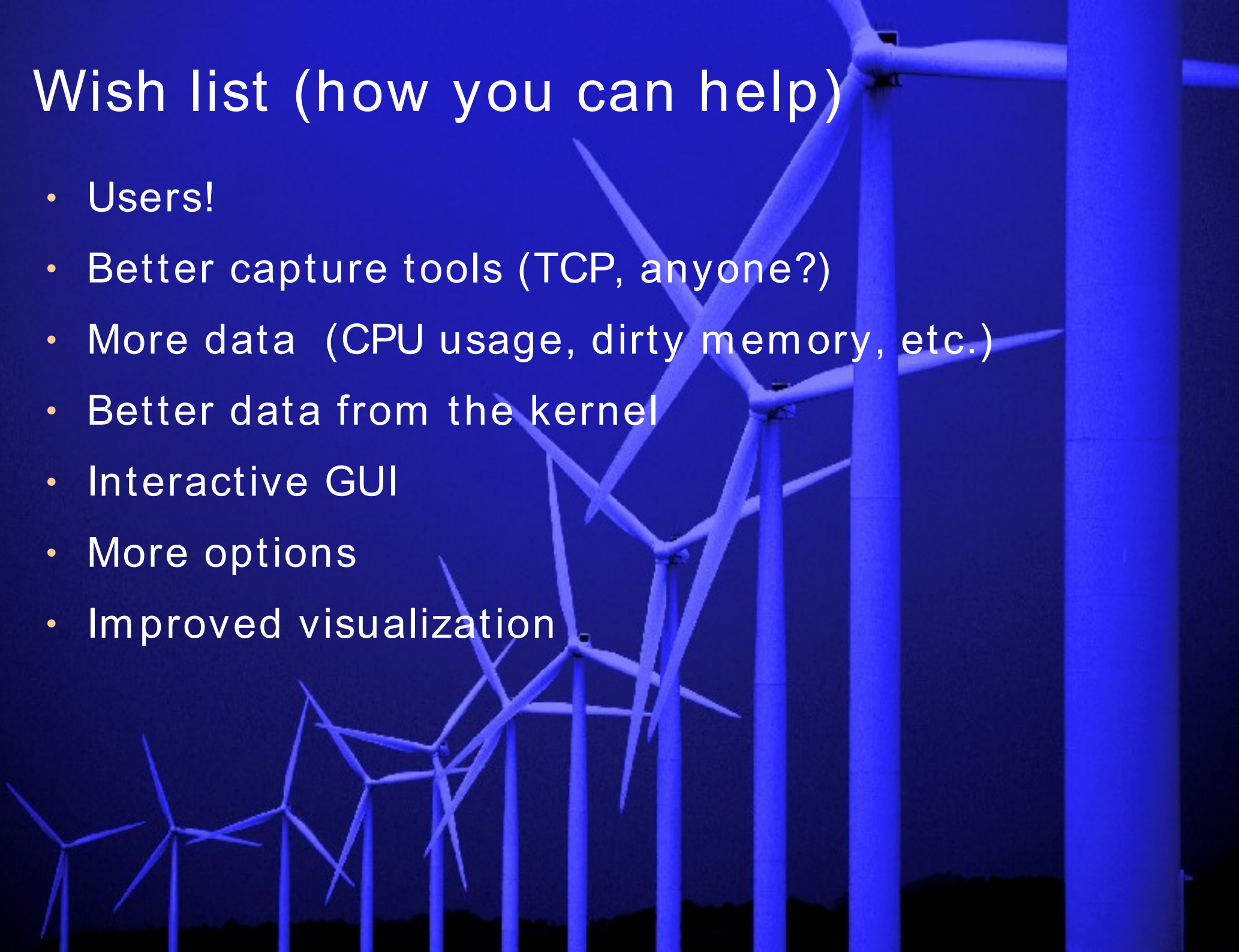
Visualization (pretty pictures)

- All smem's output can be converted to interactive pie and bar charts
- Can be saved as EPS, PNG, SVG, JPG, etc.



Wish list (how you can help)

- Users!
- Better capture tools (TCP, anyone?)
- More data (CPU usage, dirty memory, etc.)
- Better data from the kernel
- Interactive GUI
- More options
- Improved visualization



Getting smem v0.1

- <http://selenic.com/smem/>

