The Need for Polite Software

John MacCormick

July 2007

premise: making consumer software more responsive or “polite” would significantly enhance users’ experience

examples of impolite software

- intensive random I/O:
  - start→programs menu
  - control panel
  - add/remove programs panel
  - launching many applications
- background tasks hogging CPU and/or disk:
  - desktop search and antivirus
  - CcmExec.exe, svchost.exe
  - most Microsoft software patches
- mysterious hangs (no I/O, network or CPU activity)
  - launching Microsoft desktop search
- anything requiring a reboot
- the 5 minutes after login

[IOHogs demo]

recent innovations greatly improve politeness

- Vista has SuperFetch, ReadyBoost, I/O cancellation, I/O priorities
  - will fix many problems with I/O hogs
- Many-core CPUs could reduce rudeness of CPU hogs
  - unless fancy new programming abstractions bite us

Grand goal: eliminate ALL unnecessary sluggishness

how can we make software more polite?

- new OS construct, “urgency”
- educate and motivate developers
- other ideas:
  - specify “responsiveness contracts” for API functions
  - automatic analysis of software politeness
  - inform user and developer of reasons for sluggishness
  - prefetch random I/O that is predictable