Origins of Distributed Systems 1: Transparency

computer networks research

- around 1977
- had
  - telnet (remote login)
  - ftp (simple file transfer)
- **what next?**

software vision

- a set of networked computers with
  - interhost *transparent boundaries*
- looks centralized

Transparent boundaries means what?

- **VM level:**
  - single virtual and real address spaces
- **ipc level:**
  - single space of pids;
  - ability to send messages to local & remote
  - processes by the SAME mechanism.
  - (transparency of host boundaries)

Transparent boundaries means what?

- **data management level:**
  - physically distributed (replicated or partitioned) relational databases which appear logically unitary.
- **application level:**
  - a collection of networked computers which appears to users & application programmers

    as a single mainframe (awk!) - enterprise computing

Why bother?
just use a mainframe!

Origins of Distributed Systems 2: microelectronics:

- microprocessors on single chip
  - $2 *manufacturing* cost , versus
  - $5M - $500M *development* cost

- desire to make a hatful of microprocessors act as a single, large, fast cpu.

Origins of Distributed Systems 3: locality of reference

- data often shows

  geographic locality of reference

  or

  functional locality of reference

- so let’s keep it where it's used

Origins of Distributed Systems 4: mainframe-as-dinosaur:

Multiplexing

- hardware architects stood on their heads to make single fast cpus
  (pipelining, pingpong memory, caching), but
- timesharing users mostly wanted their own private machines, so
- VM/370 (CP/CMS) multiplexed many small virtual cpus
  into one expensive big one.
- Cascaded complexities! KISS.

Origins of Distributed Systems 5: mainframe-as-dinosaur

- **monopoly vs. competition:**

monopoly vs. competition:

- **mainframes:**
- proprietary
  - hardware architecture (VAX, /370)
  - operating system (Vax VMS, VM/370, MVS)
  - networking architecture (DECNet, SNA)
- hence quasi-monopolistic
  - ("locked-in customers")
- hence expensive

monopoly vs. competition:

- **PCs:**
  - commodity microprocessors
    - (Intel n86, M 68000, PowerPC)
  - commodity operating systems
    - (Windows, unix™)
  - hence cheaper cycles ($O(10 - 10^2)$!)
    - (plus easier-ported applications
  - more & smarter software development)

monopoly vs. competition:

- PCs:
  - lower hardware cost permitted:
    - cheaper computing
    - high-res graphical interfaces
    - plentiful cpu cycles, Mp, Ms (but not I/O)
  - which permitted point-and-grunt user interfaces
  - (MacOS, Windows, X-windows)

Hardware models for Distributed Systems:

Hardware Models:

- **1. Multiprocessor**
  - shared Mp,
  - common clock
  - fast, error-free ipc over small, fast busses

Hardware Models:

- **2. distributed system**
  - no physically shared Mp
  - no common clock
  - slow, long-latency,expensive , error-ridden ipc over
  - long, slow LANs & WANs.

Hardware Models:

- 3. Future distributed systems?
  - no physically shared Mp
  - no common clock
  - Fast, cheap error-free ipc over
  - long, fast LANs & WANs.(OPTICAL FIBRE!)
    - but LATENCY will remain long , alas
    - (speed of photons or electrons in fibre or wire) .

Evolution of commercial distributed systems: