OS Foundations

Peter J. Denning

October 4, 2015

Our Story

OS Principles began emerging 1960,

Grew across many generations of technologies,

And left a rich heritage in the minds, hearts, and souls of all who use computers.

Our Story

Timelines

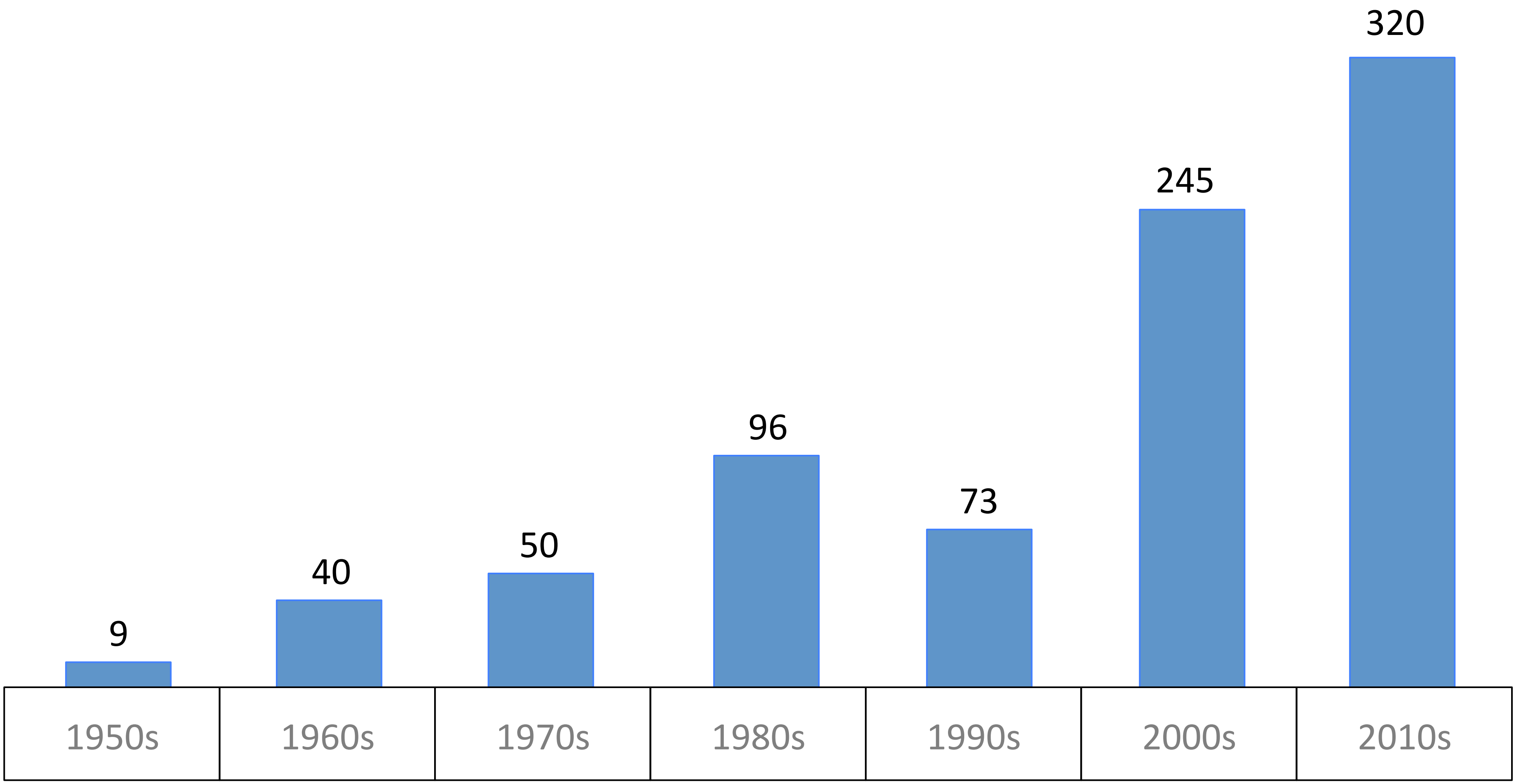
Personal example

Research Lessons

OS a force on all of computing

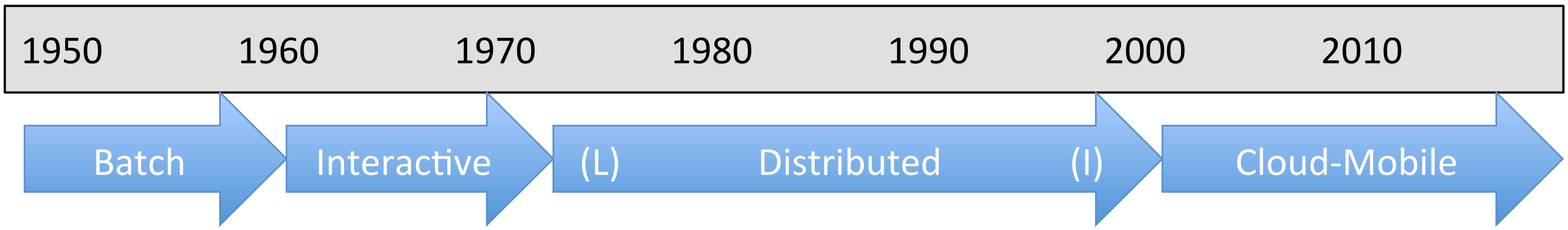
We will be surfing …

Source: public domain internet



Number of new OS’s per decade (Wikipedia)

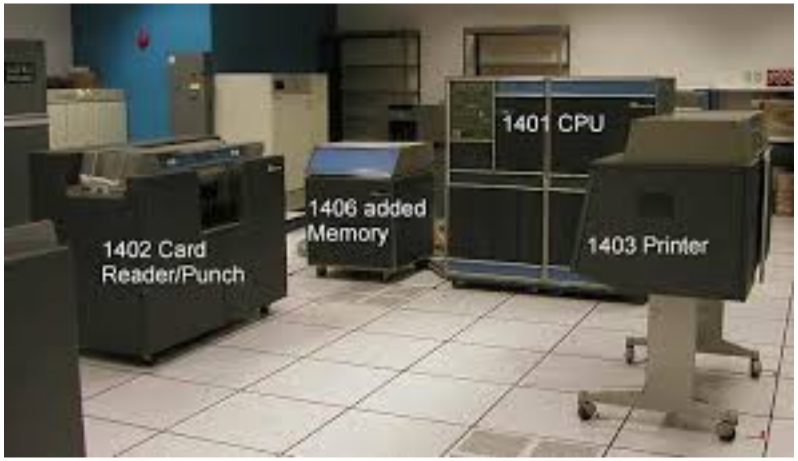
Eras of Operating Systems



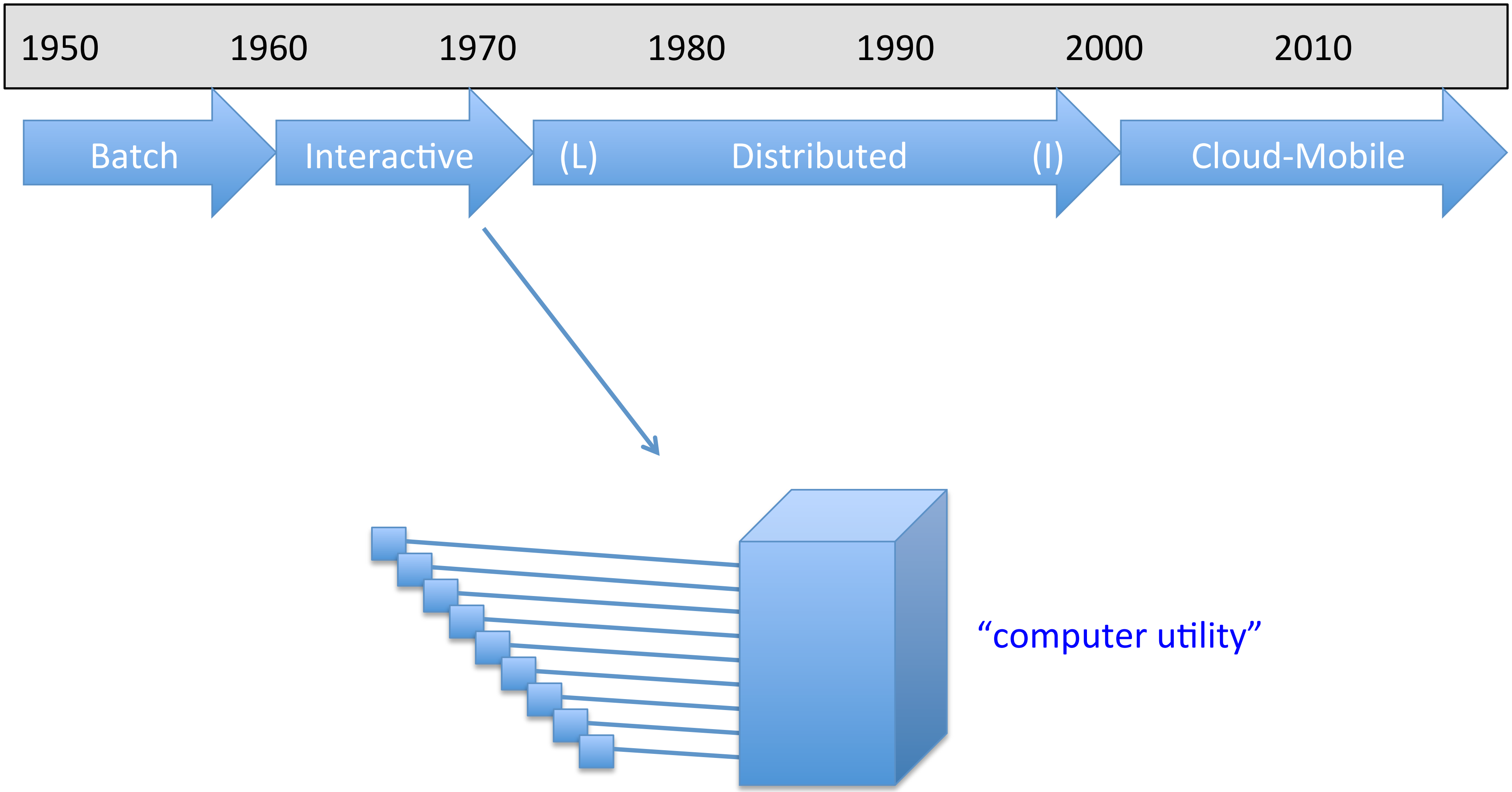
Eras of Operating Systems

1950 1960 1970 1980 1990 2000 2010

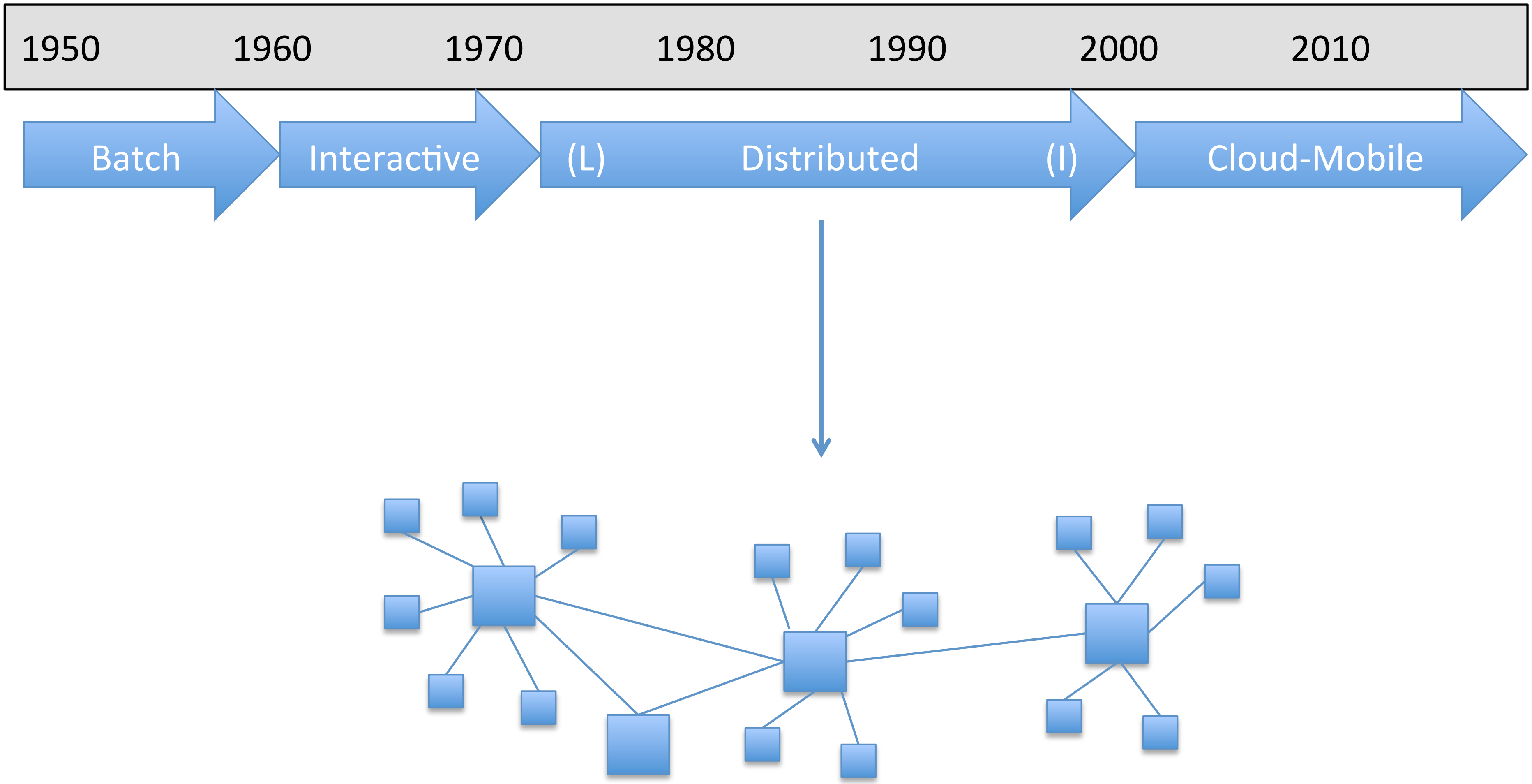
Batch → Interactive → (L) Distributed (I) → Cloud-Mobile



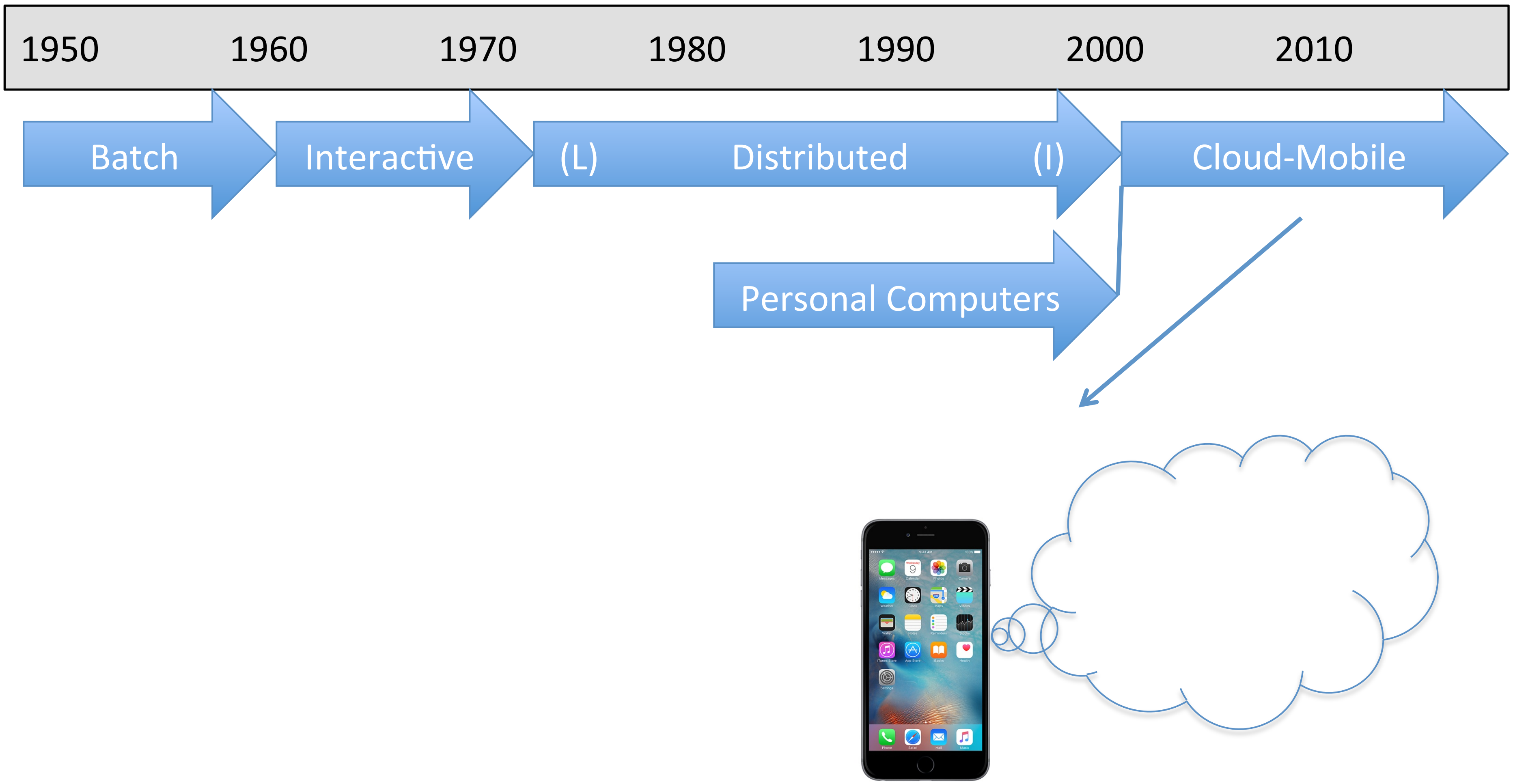
Eras of Operating Systems



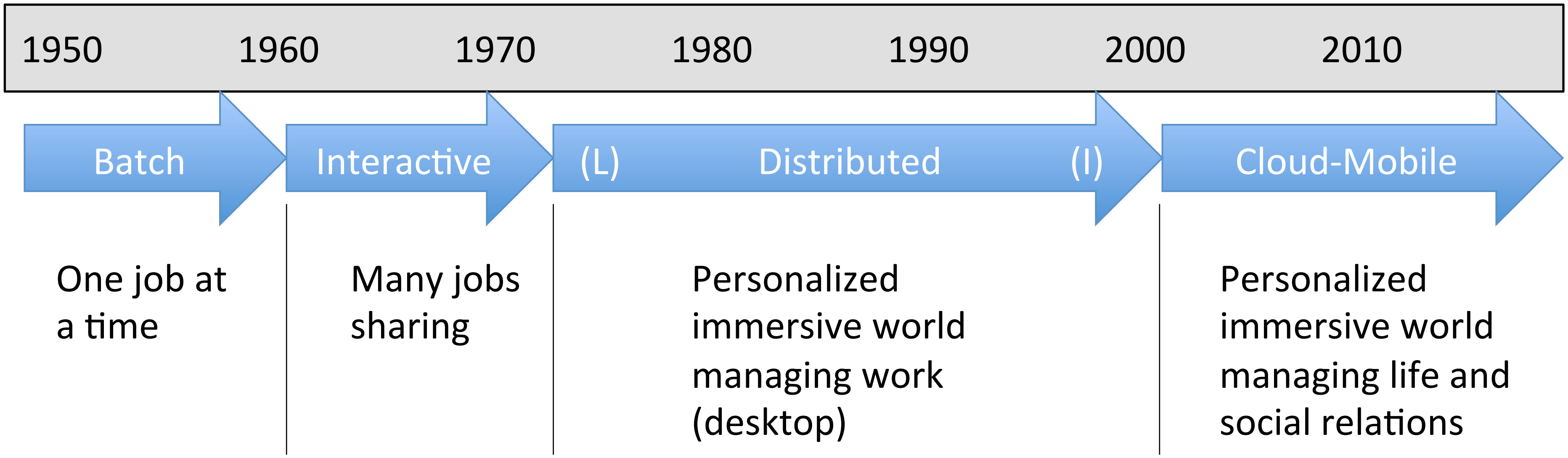
Eras of Operating Systems



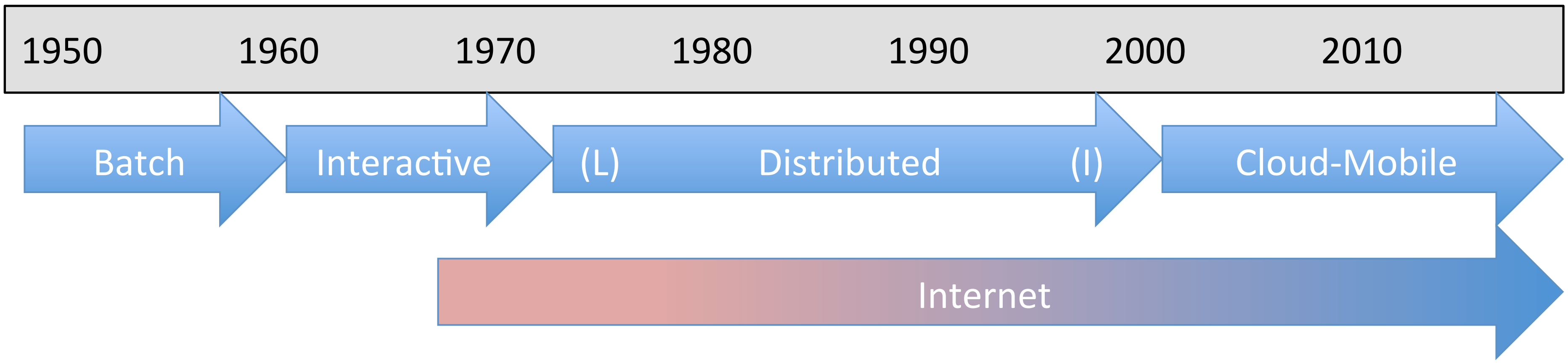
Eras of Operating Systems



Eras of Operating Systems



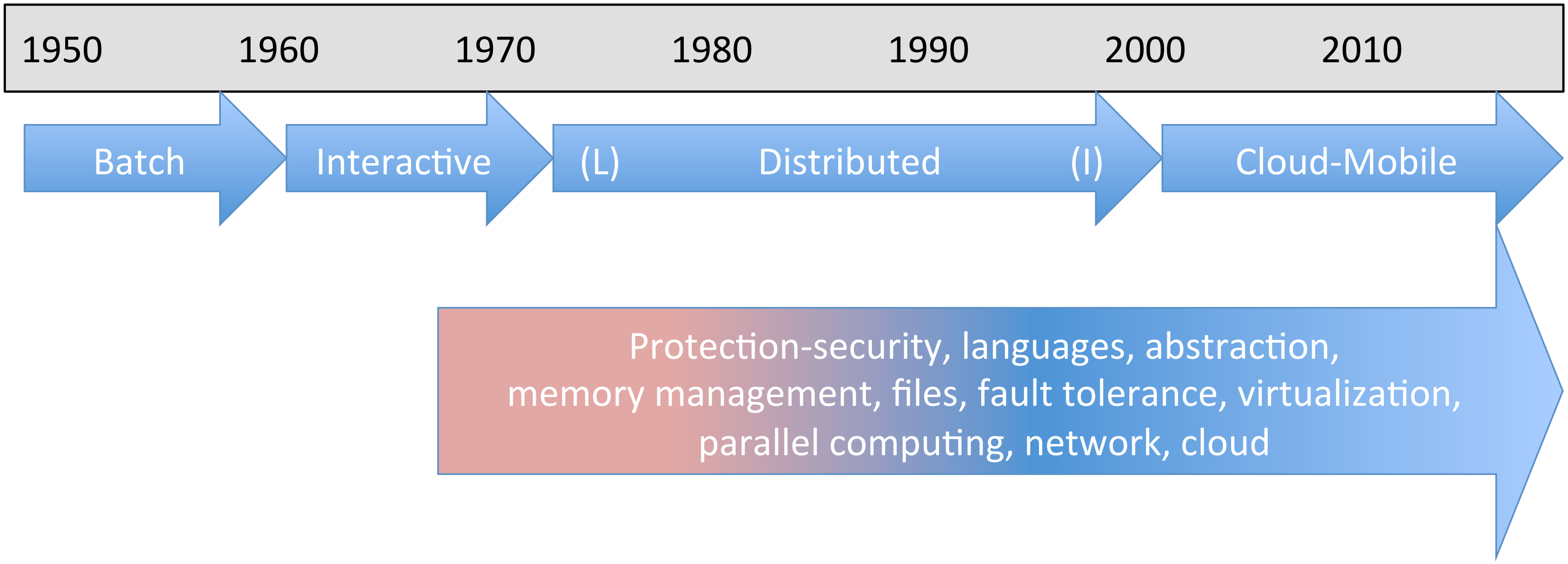
Eras of Operating Systems



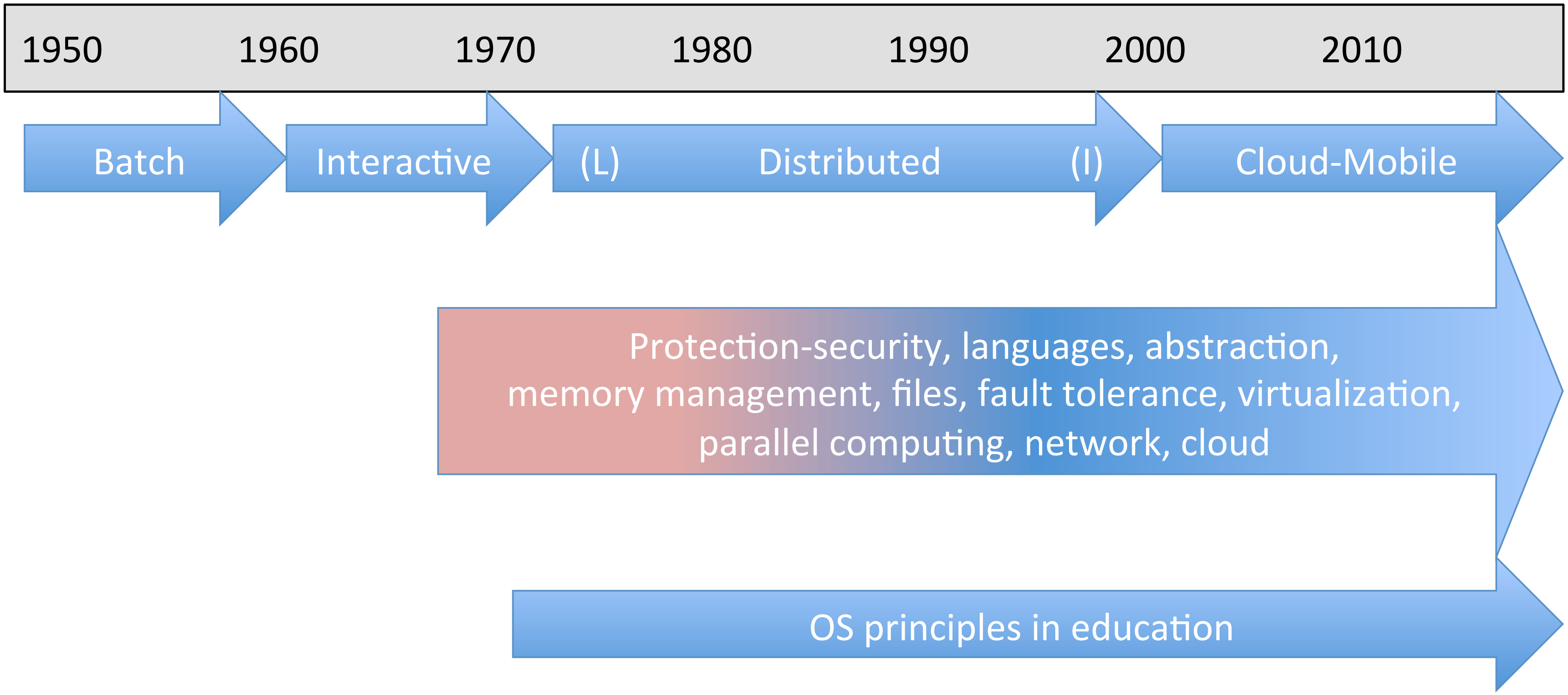
"OS interfaces with":
TELNET
FTP
SMTP
RLOGIN
RCOPY

"OS integrates with":
Protocol software
IPC, RPC
Daemon processes
Client-server, X-windows
Hyperlink, URL
Browser
Search

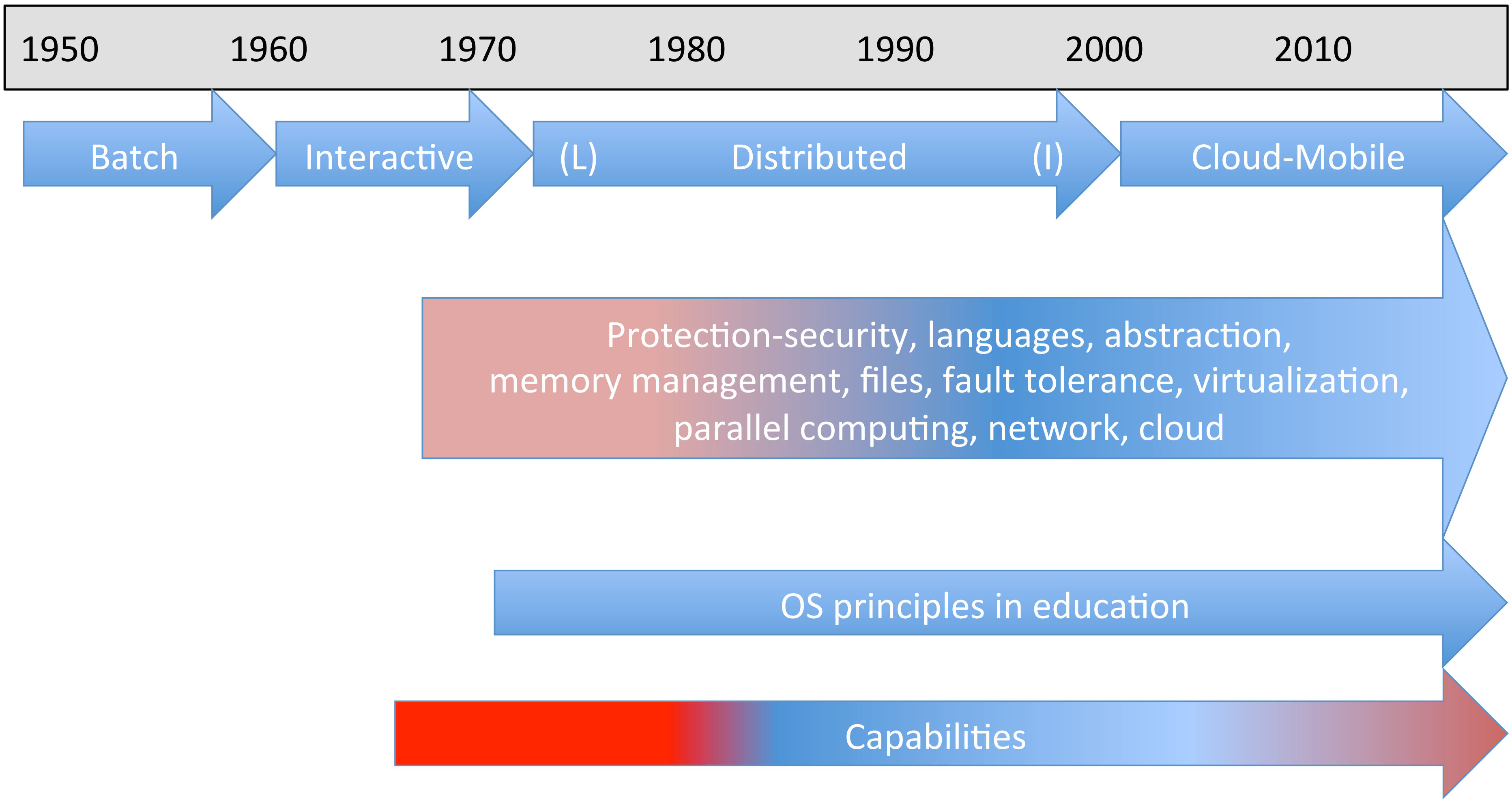
Eras of Operating Systems



Eras of Operating Systems



Eras of Operating Systems



Development of Principles

What is a computing principle?

Development of Principles

What is a computing principle?

Law

Development of Principles

What is a computing principle?

Law

c(t) = min(a(t), s(t)+I)

M = (spacetime) x (throughput)

Mean Value Equations

Locality

Development of Principles

What is a computing principle?

Law

Statement of Design Wisdom

Development of Principles

What is a computing principle?

Law

Statement of Design Wisdom

Information hiding

Levels of abstraction

Development of Principles

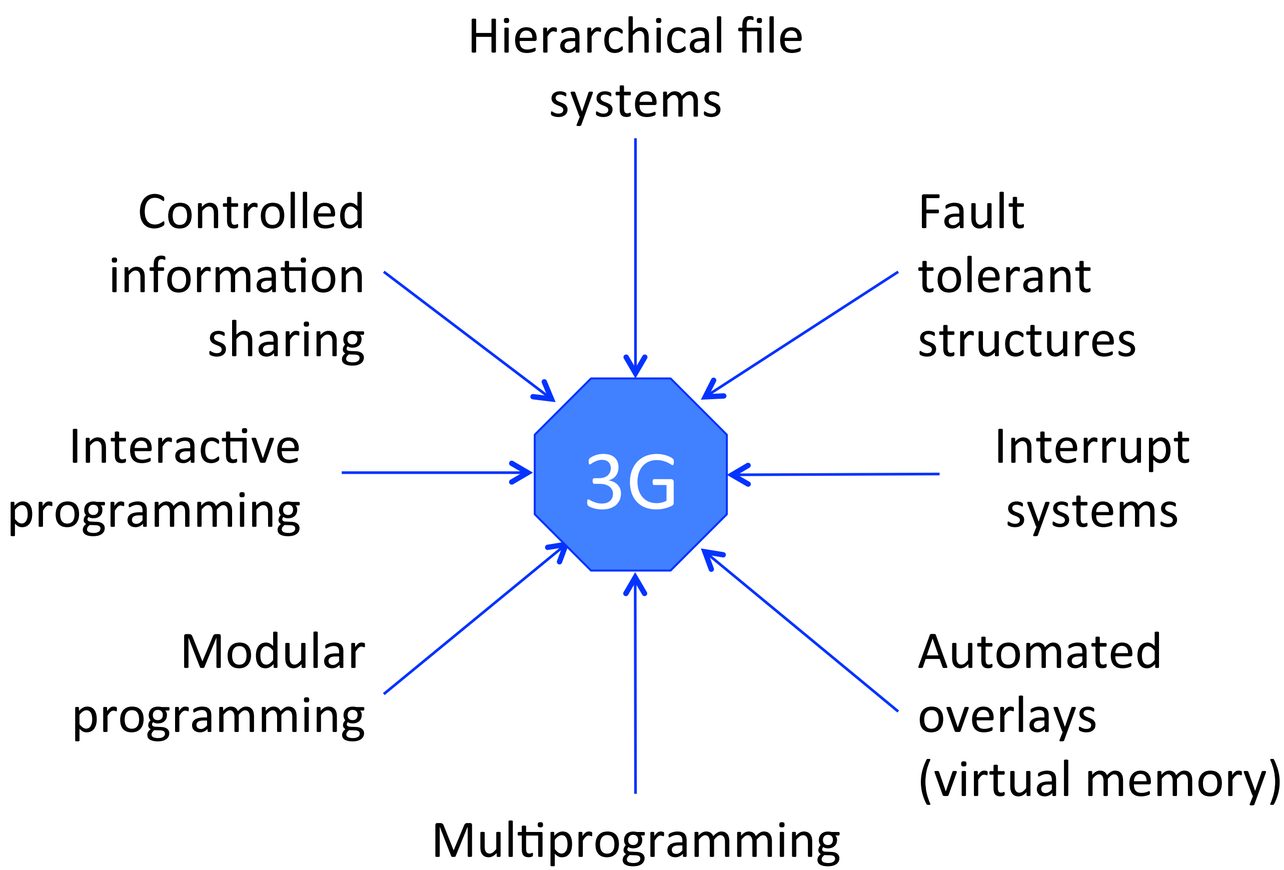
What is a computing principle?

Law

Statement of Design Wisdom

Cosmic = timeless + spaceless (Jim Gray)

Eight programming support objectives added by 1965 seeded the research



+HLLs for programming the OS

Principles govern only the recurrences part of our story.

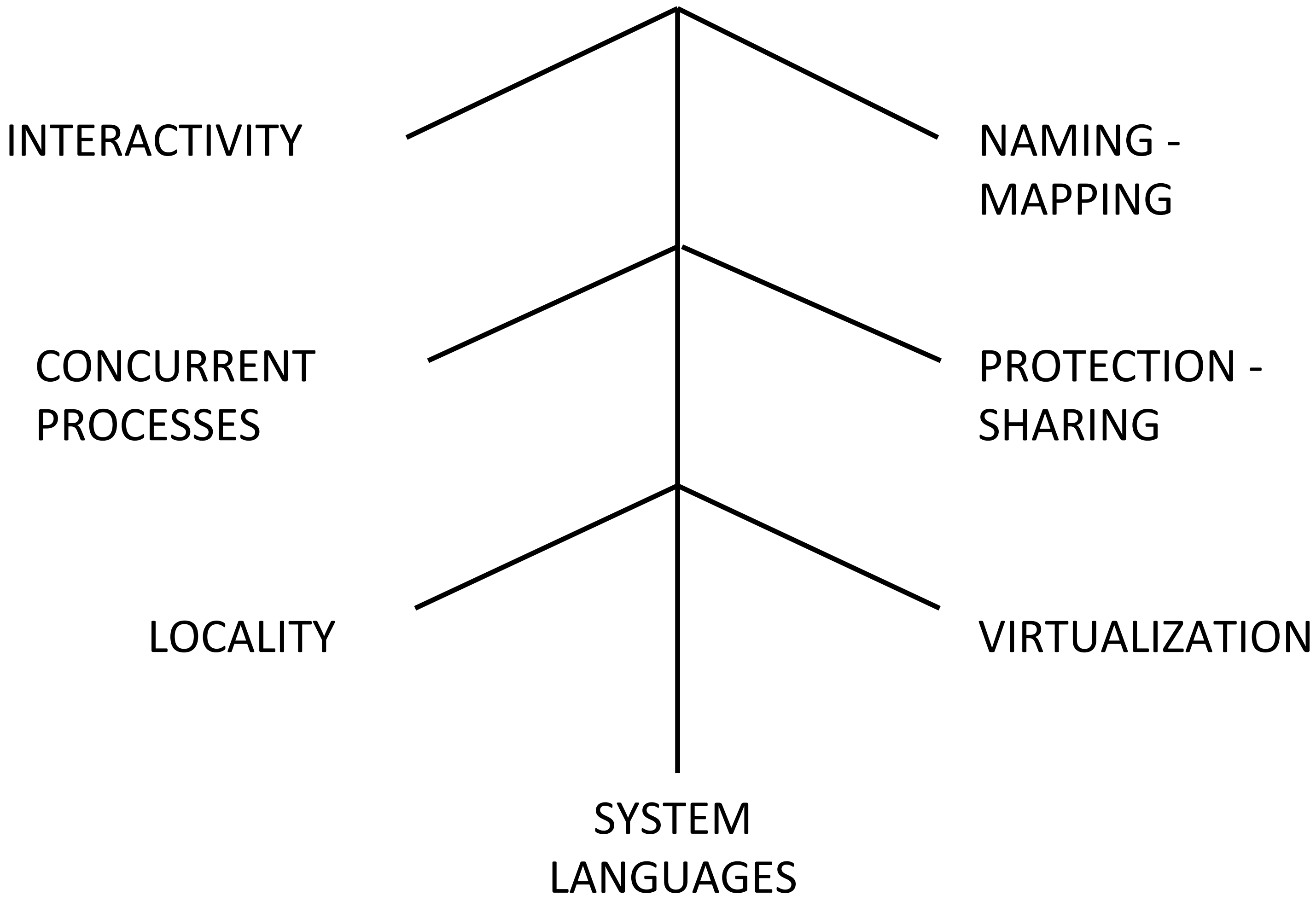
Accidents and unforeseen contingencies constantly appear.

We respond to them:

With bug fixes, patches, new designs, new apps.

With research seeking greater understanding and occasionally opening new insights and new principles.

OS Principles into CS



OS Principles into CS

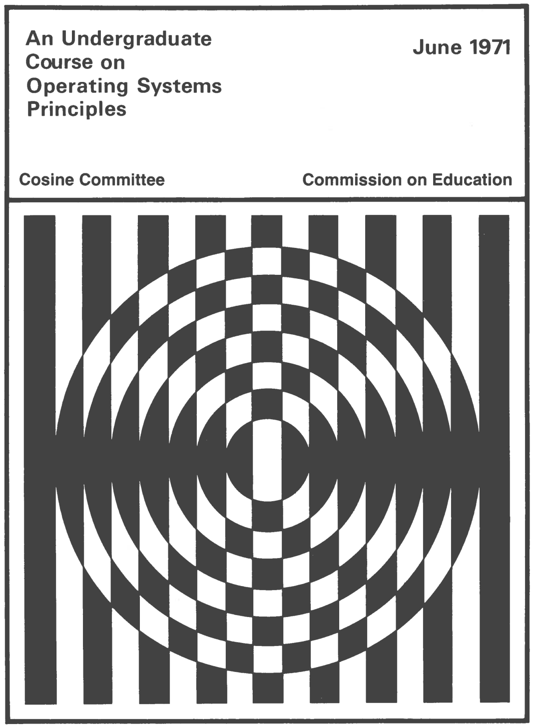
In all,

13 of 41 CS Principles (30%)

Source: greatprinciples.org

OS Principles into CS

- First non-math core course 1971
- Expanded definition
  of core to include systems
- Unchallenged for 44 years



Two Cosmic Principles Revealed in Memory Management

Locality

Location independent addressing

Emerged from virtual memory research

Originally seeking to automate overlays

Locality Principle

Big Adventure

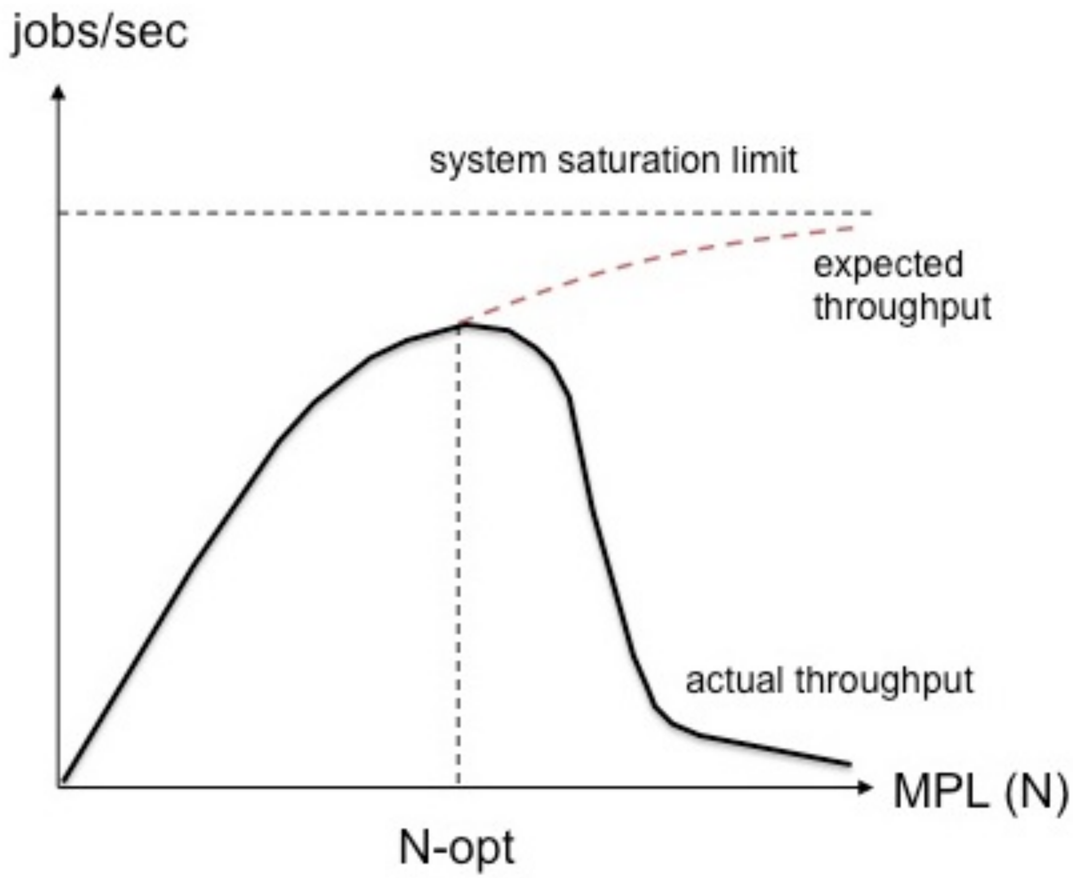
I became involved at Project MAC 1965

Many people involved (thousands!)

First motivations ca. 1965 performance related:

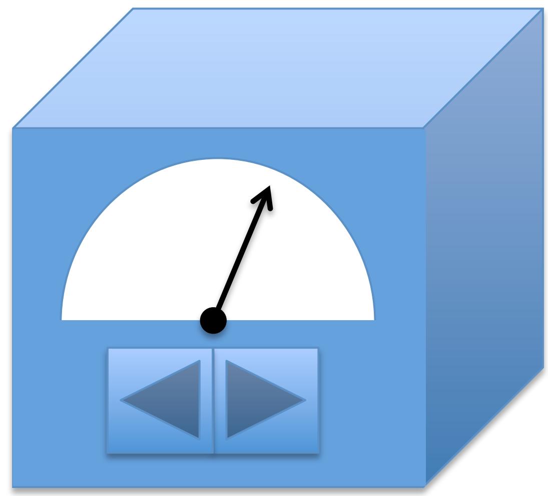
- Performance of virtual memory
- Multiprogramming
- Thrashing

Thrashing

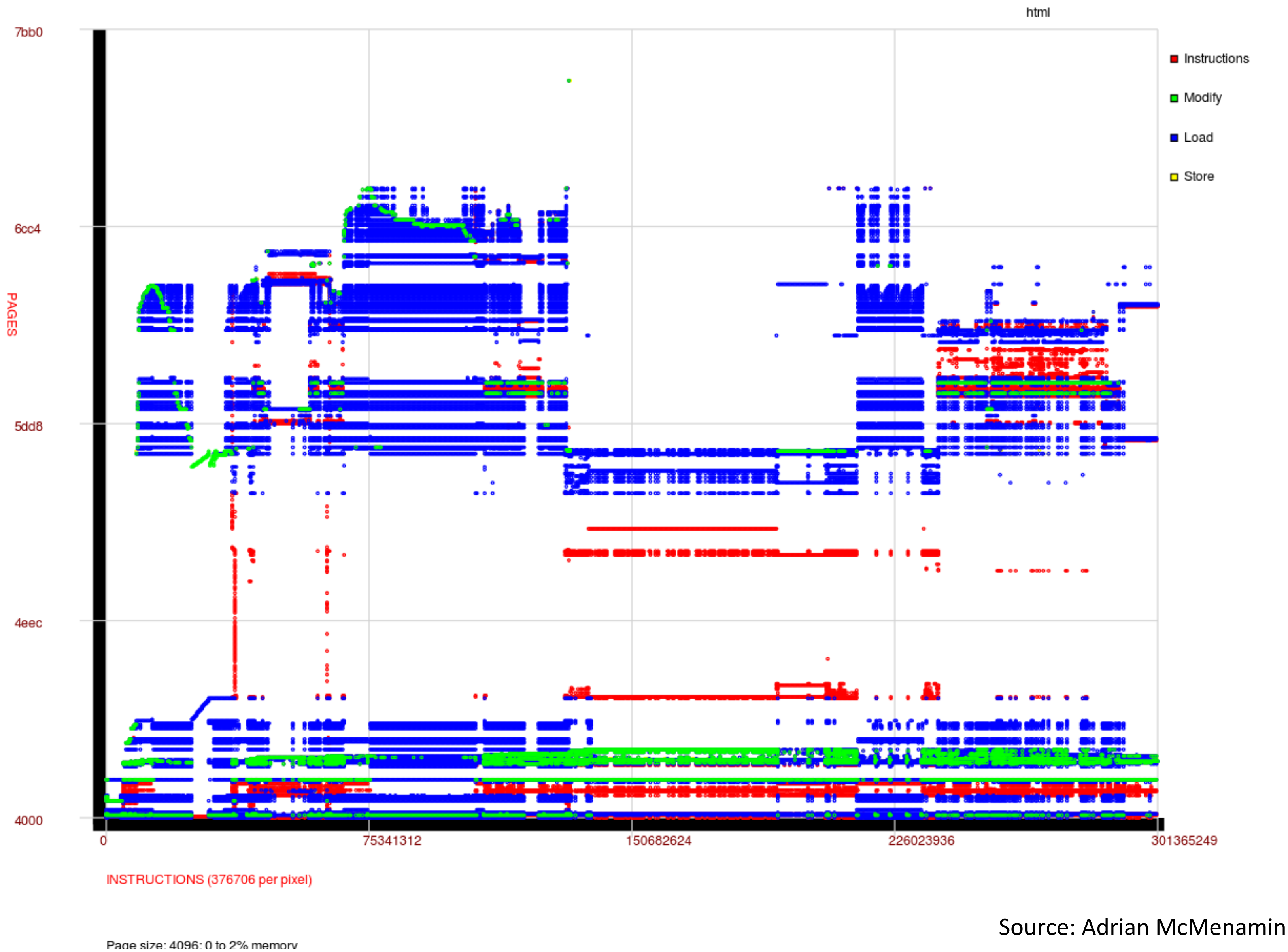

jobs/sec
system saturation limit
expected
throughput
actual throughput
MPL (N)
N-opt

Saltzer's Challenge

Tune one parameter to lock in optimal performance



Reference Map – Initial Intuitions



Source: Adrian McMenamin

Key insights:

- Temporal and spatial clustering (Belady, Denning 1966)
- Working sets (Denning 1966)
- Reference maps (IBM ca 1969)
- Optimality principle (Belady 1966, Prieve and Fabry 1976, Gray 1995)

Working set

Pages used in previous
virtual time window
of size T

Vertical slice in
page reference map

What We Have Learned

WS = locality set most of the time

WS policy near optimal (VMIN)

Economical implementations (WSCLOCK)

Prevents thrashing

Answers Saltzer's challenge

Locality Principle

Initial intuitions confirmed

All computations display locality (empirical)

All computations must display locality (theory)

Harnessing locality always pays off

- caching
- parallelizing
- performance
- no-thrashing

Location Independent Addressing

Key insights:

- Paging (U Manchester 1949)
- Virtual v. real address (ca 1959 Kilburn and Fotheringham)
- Segmentation in universal hierarchical address space (Dennis 1965)

From which flowed:

- Dynamic mapping virtual to real via page table
- Demand paging
- Replacement algorithms
- MMU and TLB mapping architecture
- Hierarchical naming systems

Huge benefits:

- Location independence
- Logical partitioning (address space isolation)
- Artificial contiguity
- Relocation
- Distributed naming authorities

Evolved into global, all-time unique addresses for digital objects anywhere in a system.

Hierarchical Internet URLs and domain names.

Now Internet is a huge virtual address space of capabilities, URLs, and DOIs with mapping via DNS and handle-servers.

Patterns I learned from OS Research

1. There is never certainty
2. Occasionally an insight charts a new direction
3. Technology inflection points may trigger avalanches
4. Searching for what works: building, experimenting, tinkering
5. Always in a social network
6. Theory follows practice

Summary

- Rich heritage from OS evolution

Summary

- Rich heritage from OS evolution

  Batch … Interactive … Immersive

Summary

- Rich heritage from OS evolution

  Batch ... Interactive ... Immersive

- Fundamental CS principles emerged from our research

Summary

- Rich heritage from OS evolution

  Batch … Interactive … Immersive

- Fundamental CS principles emerged from our research
- OS drove major change to CS curriculum

Summary

- Rich heritage from OS evolution

  Batch … Interactive … Immersive

- Fundamental CS principles emerged from our research
- OS drove major change to CS curriculum
- OS research a unique blend of engineering, experimenting, and modeling

Challenges Ahead

Automation of knowledge work – end of "knowledge age"

Network: space of social power, action, identities

Security and privacy in the Internet of Things

Size, complexity, scale of systems

Integrating with bio and nano tech

End of this story