# Identifying Stable File Access Patterns

Purvi Shah, University of Houston
Jehan-François Pâris, University of Houston
Ahmed Amer, University of Pittsburgh
Darrell D. E. Long, U. C. Santa Cruz

# The Problem

- *Disk drive capacities* double every year
- *Access times* have only decreased by a factor of 3 over the last 25 years
- *Cannot keep up* with increased I/O traffic resulting from faster CPUs
- Two most used solutions are
  - *Caching*
  - *Prefetching*

# Caching

- Keeps recently accessed data in memory

- Used by nearly all systems

- Scale boosted by availability of cheaper RAM

  – Can now cache entire files

- Does not work well at *server level* in the presence of a *larger client cache*

# File Prefetching

- *Predictive prefetching:*
  - Tries to predict which files will be accessed next in order to fetch them before they are needed

- *Implicit prefetching:*
  - Tries to build clusters of *related files* that can be fetched in a *single I/O operation*
  - Assumes existence of stable long-lived file access patterns

# First Stable Successor

- New file predictor that identifies *stable access patterns*

- Keeps track of the successor of each file

- Predicts that file $Y$ will *always* be the successor of file $X$ once it has detected $m$ successive accesses to file $Y$, each immediately following an access to file $X$

  *XZ … XY … XU … XY … XY … XY … X?*

  If $m = 3$, predicts that $Y$ is successor of $X$

- Never alters this prediction

# Performance Criteria

- Two traditional metrics
  - *Success-per-prediction*
  - *Success-per-reference*
- Neither of them capture whole performance of a file access predictor
- Use instead effective miss ratio

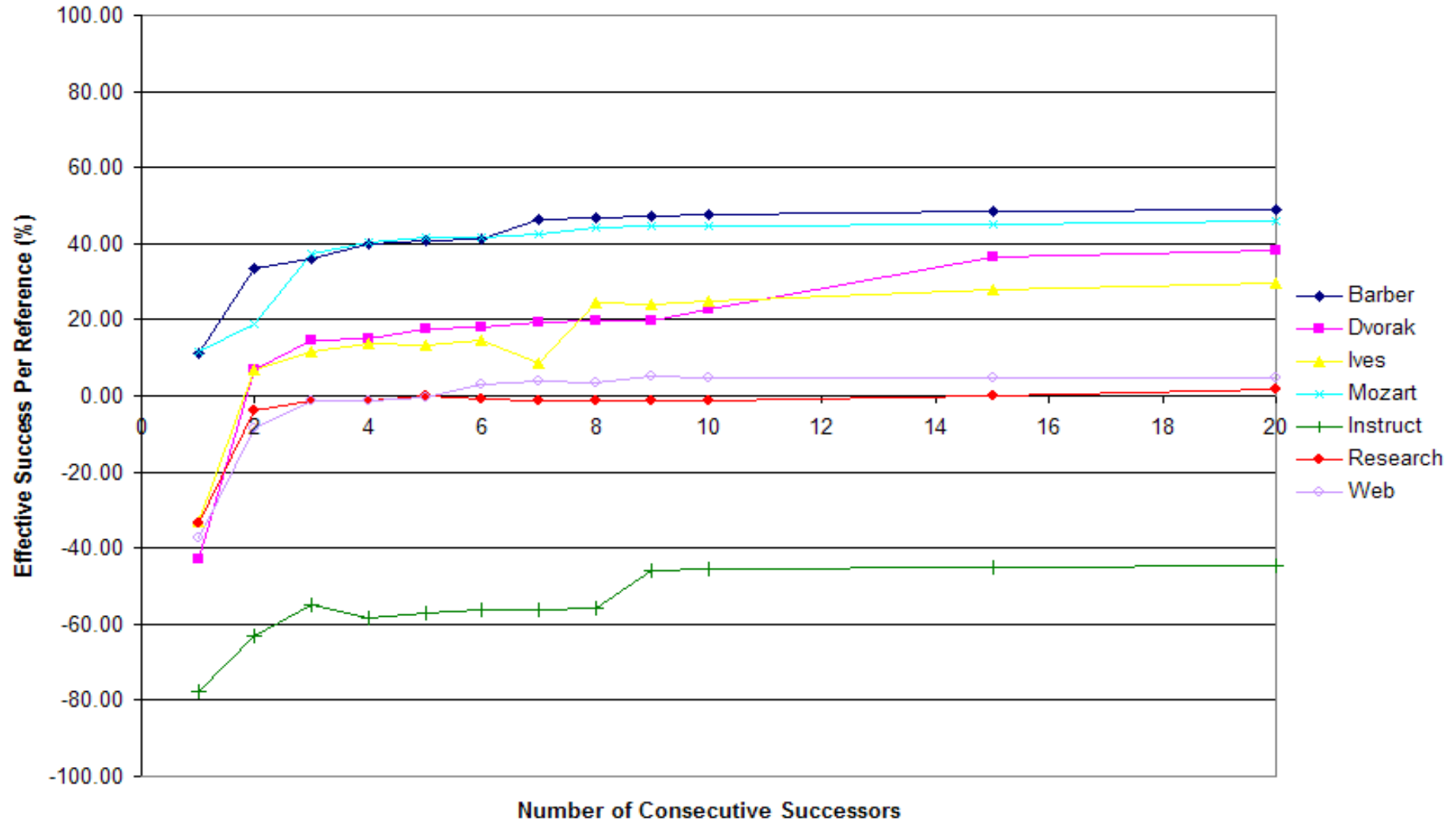$$\frac{N_{corr} - \alpha N_{incorr}}{N_{ref}}$$

where $0 \le \alpha \le 1$ represents the relative cost of a wrong prediction
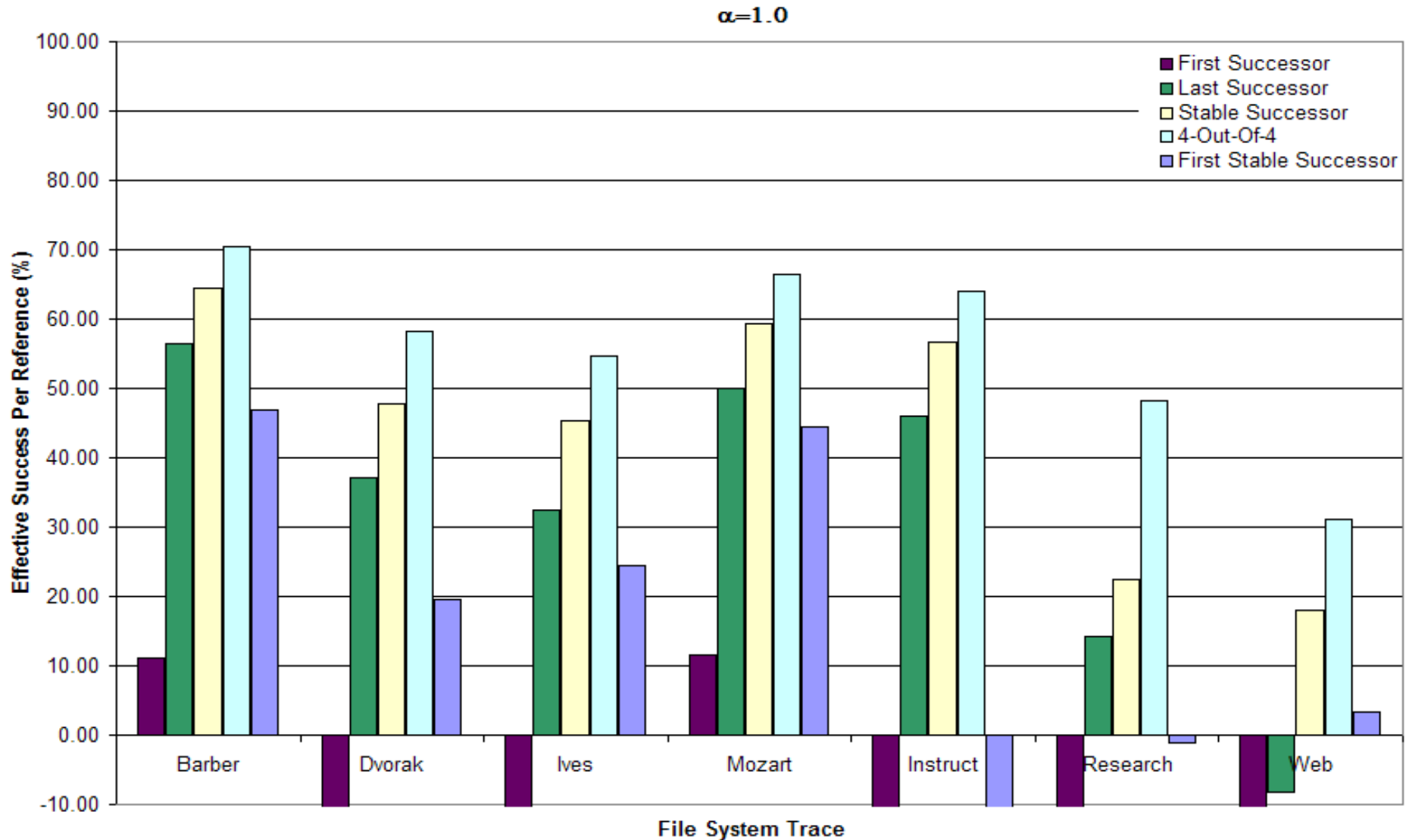
# Experimental Setup

- We evaluated the performance of our FSS predictor by simulating its operation on two set of file system traces
  - Four traces collected at CMU
    - Mozart, Ives, Dvorak and Barber
    - Cover a period of about a year
  - Three traces collected at Berkeley
    - Web, Instruct and Research
    - Recorded over approximately three month period

# Effect of Successor History Length (α=1.0)



α=1.0

Effective Success Per Reference (%) vs Number of Consecutive Successors

Legend: Barber, Dvorak, Ives, Mozart, Instruct, Research, Web

# How FSS compares with other predictors ($\alpha$=1.0)

# Conclusions

- First Stable Successor (FSS) offers a practical tool for identifying stable access patterns

- Enough of them to make implicit file prefetching a worthwhile proposition

- Many, if not most, of these patterns appear to persist over several weeks if not a whole year

# Thank-you!

NASA/IEEE MSST 2004
12th NASA Goddard/21st IEEE Conference on
Mass Storage Systems & Technologies
The Inn and Conference Center
University of Maryland University College
Adelphi MD USA
April 13-16, 2004