On the Smoothness of Paging Algorithms

Jan Reineke and Alejandro Salinger

Technische Universität Dortmund
November 17, 2016
Motivation: Real-time Systems

Controllers must finish their tasks within given time bounds.

→ Need to determine Worst-Case Response Times (WCRT)
Influence of Caches on Execution Times

\[ x = a + b; \]

\[
\begin{align*}
    \text{LOAD} & \quad r2, _a \\
    \text{LOAD} & \quad r1, _b \\
    \text{ADD} & \quad r3, r2, r1 
\end{align*}
\]

PowerPC 755

Execution Time (Clock Cycles)
Basics: Caching/Paging

Cache

Slow memory

\[ \sigma = \cdots p_6 p_3 p_2 p_4 p_4 p_2 p_{10} p_{11} p_5 p_4 \cdots \]

Is \( p_i \) in the cache?

- Yes  Hit

- No  Fault (miss)

Fetch \( p_i \) from slow memory, evict one page from cache

→ Replacement policy determines page to evict
→ Access sequence + policy determine cache state
Why are Caches a Challenge?

1. Input-dependent memory accesses
2. Interference due to preempted tasks
3. Interference due to co-running tasks
Two Approaches to Timing Analysis

1. Static Analysis:
   
   How does uncertainty about memory accesses affect uncertainty about number of faults?

2. Measurement-based Analysis:
   
   How representative are measurements on a subset of the possible cases?
How does uncertainty about memory accesses affect uncertainty about execution time?

Quantify uncertainty by edit distance

number of faults

$A(\sigma')$

$A(\sigma)$

access sequences

$\sigma$  $\sigma'$
Definition of Smoothness

\( \text{dist}(\sigma, \sigma') = \text{edit distance between } \sigma \text{ and } \sigma' \)

An online algorithm \( A \) is \((\alpha, \beta, \delta)\)-smooth if for all \( \sigma, \sigma' \) with \( \text{dist}(\sigma, \sigma') \leq \delta \)

\[
A(\sigma') \leq \alpha \cdot A(\sigma) + \beta
\]

For generic \( \delta \):

An online algorithm \( A \) is \((\alpha, \beta)\)-smooth if for all \( \sigma, \sigma' \)

\[
A(\sigma') \leq \alpha(\delta) \cdot A(\sigma) + \beta(\delta)
\]

where \( \alpha \) and \( \beta \) are functions and \( \text{dist}(\sigma, \sigma') \leq \delta \)

We call \( A \) smooth if it is \((1, \beta, 1)\)-smooth for some \( \beta \).
Key Questions

• How smooth are known paging algorithms?

• Are there fundamental bounds on the smoothness of paging algorithms?

• Are smoothness and high performance contradictory goals?

• Can randomization help?
Deterministic Replacement Policies / Paging Algorithms

Online Algorithms:
- LRU: Least-Recently-Used
- FIFO: First-In-First-Out
- FWF: Flush-When-Full
- ...

Offline Algorithm:
- FITF: Furthest-In-The-Future (OPT, LFD, Belady’s)
Smoothness of LRU

\[ \sigma = \ldots a \ b \ c \ d \ a \ b \ c \ d \ \ldots \]

\[ \sigma' = \ldots a \ b \ c \ d \ p \ a \ b \ c \ d \ \ldots \]

\[ \text{dist}(\sigma, \sigma') = 1 \]

\[ LRU(\sigma) \quad [d \ c \ b \ a] \rightarrow [d \ c \ b \ a] \rightarrow [a \ d \ c \ b] \rightarrow [b \ a \ d \ c] \]

\[ LRU(\sigma') \quad [d \ c \ b \ a] \rightarrow [p \ d \ c \ b] \rightarrow [a \ p \ d \ c] \rightarrow [b \ a \ p \ d] \]

miss \quad miss \quad miss

\[ \rightarrow [c \ b \ a \ d] \rightarrow [d \ c \ b \ a] \]

\[ LRU(\sigma') = LRU(\sigma) + 5 \]
Smoothness of LRU

\[ \forall \sigma, \sigma' \text{ with } dist(\sigma, \sigma') = 1 \]

\[ LRU(\sigma') \leq LRU(\sigma) + (k + 1) \]

Proof sketch

- Age of page: number of distinct requests since last request
- A request is a miss if age is \( \geq k \)
- At all times at most \( k \) pages have age \( < k \)
- New page \( p \) can increase age of at most \( k \) requests from \( k - 1 \) to \( k \)

\[ \Rightarrow \text{At most } (k + 1) \text{ extra misses} \]
What about $\delta > 1$?

For any replacement policy $A$:
If $A$ is $(1,c,1)$-smooth, then $A$ is $(1,\delta c)$-smooth

$$LRU(\sigma') \leq LRU(\sigma) + (k + 1) \text{ for } \sigma, \sigma' \text{ with } dist(\sigma, \sigma') = 1$$

$$LRU(\sigma') \leq LRU(\sigma) + \delta(k + 1) \text{ for } \sigma, \sigma' \text{ with } dist(\sigma, \sigma') \leq \delta$$

$LRU$ is $(1, \delta(k + 1))$-smooth
FIFO is not smooth

There exist $\sigma$ and $\sigma'$ with $\text{dist}(\sigma, \sigma') = 1$ such that caches of $\text{FIFO}(\sigma)$ and $\text{FIFO}(\sigma')$ are reversed.

**FIFO($\sigma$)**

Initial state: [a b c d]

1. e
2. [e a b c] (miss)
3. a
4. [e a b c] (miss)
5. [e a b c] (miss)

**FIFO($\sigma'$)**

Initial state: [d c b a]

1. e
2. [e d c b] (miss)
3. a
4. [a e d c] (miss)
5. [b a e d]

...
Lower Bounds (1 of 2)

• An algorithm is *demand paging* if it only evicts pages when needed
  – e.g., LRU, FIFO

No deterministic, *demand-paging* algorithm is better than

\[(1, \delta(k + 1))-\text{smooth}\]

• But not all algorithms are demand paging:

  How about *competitive* algorithms?
Competitive Analysis

• \( A(\sigma) \): number of misses of A on sequence \( \sigma \)
• An online algorithm A is \( r \)-competitive if for all \( \sigma \)

\[
A(\sigma) \leq r \cdot OPT(\sigma) + \beta
\]

• Minimum such \( r \) is A’s competitive ratio \( CR(A) \)
• \( CR(LRU) = CR(FIFO) = CR(FWF) = k \)
• Randomized algorithms can achieve CR in \( O(\log k) \)
Lower Bounds (2 of 2)

• An algorithm is *demand paging* if it only evicts pages when needed
  – e.g., LRU, FIFO

No deterministic, **demand-paging** algorithm is better than

\[(1, \delta(k + 1))-\text{smooth}\]

• But not all algorithms are demand paging
  – e.g., FWF

No deterministic, **competitive** algorithm is better than

\[(1, \delta(k + 1))-\text{smooth}\]
Deterministic Algorithms

- Smoothness
- Deterministic strongly-competitive
- c-competitive
- FIFO
- FWF
- LRU
- OPT

18

1

k

c

Competitiveness
Can randomization help?
Randomized Replacement Policies

- **RAND**: Evict page chosen uniformly at random
- **MARK**: Evict only “unmarked” pages
- **PARTITION, EQUITABLE**: define state probability distribution based on OPT’s cache contents
- **Evict-On-Access**: like RAND, but evict on hits too!
## Randomized Replacement Policies

| Algorithm     | Competitive ratio | Smoothness                     |
|---------------|-------------------|-------------------------------|
| RANDOM        | $k$               | $(1, \delta(k + 1))$          |
| MARK          | $2H_k - 1$        | $(\Theta(H_k), \beta)$       |
| PARTITION     | $H_k$             | $(1 + \epsilon, \beta, 1)$   | $(H_k, 2H_k)$                  |
| EQUITABLE     | $H_k$             | $(1 + \epsilon, \beta, 1)$   | $(H_k, 2H_k)$                  |
| EOA           | $\infty$          | $\left(1, \left(1 + \frac{k}{2k - 1}\right)\delta\right)$ |

\[
H_k = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}
\]

No randomized, demand-paging algorithm is better than

\[
\left(1, H_k + \frac{1}{k}, 1\right)\text{-smooth}
\]

No randomized, strongly-competitive algorithm is better than

\[
(1, \delta H_k)\text{-smooth}
\]
**RANDOM** is \((1, \delta(k + 1))\)-smooth

- Bound extra misses by distance between distributions

\[ D_1 \]

\[
\begin{align*}
[a \ b \ c] & \rightarrow 1/3 \\
[a \ b \ d] & \rightarrow 1/3 \\
[a \ c \ d] & \rightarrow 1/3 \\
\end{align*}
\]

\[ D_2 \]

\[
\begin{align*}
[a \ b \ c] & \rightarrow 2/3 \\
[a \ b \ e] & \rightarrow 1/3 \\
\end{align*}
\]

\[
\Delta(D_1, D_2) = \min_{\alpha} \sum_{s_1 \in D_1, s_2 \in D_2} \alpha(s_1, s_2) \cdot \text{cost}(s_1, s_2)
\]

\[
\text{cost}(s_1, s_2) = kH_c \quad \text{where } c = |s_1 \setminus s_2|
\]

2 Claims:

\[ D \text{ and } D' \text{ are distributions resulting from serving } \rho \text{ and } \rho' \text{ with } \text{dist}(\rho, \rho') = 1. \text{ Then, } \Delta(D, D') \leq k \]

\[ \forall \sigma, \text{RAND}(D_1, \sigma) - \text{RAND}(D_2, \sigma) \leq \Delta(D_1, D_2) \]
Randomized Algorithms

Smoothness

$(c, \chi)$

$(k, \gamma_2)$

$(O(H_k), \nu_1)$

$(1, \delta \beta)$

$(1, \delta(k + 1))$

$(1, \delta H_k)$

$OPT$

$MARK$

$PARTITION EQUITABLE$

$SMOOTH$

$Rand. strongly-comp.$

$FWF$

$RANDOM, LRU$

$c$-competitive

EOA

Competitiveness

$1, H_k, 2H_k - 1, k, c, \infty$
Can we design smoother algorithms?
Smoothed-LRU

- Recall the age of a page in LRU’s cache: hit if age < $k$
- Idea: smooth this transition

$k = 8$

$i = 4$
Smoothing-LRU

- Smoothed-LRU is $(1, \delta \cdot \min\left(\frac{2k-1}{2i+1} + 1, \frac{k+i-1}{2i+1} + 2\right))$-smooth

- $i = 0 \rightarrow$ as smooth as LRU
- $i = k-1 \rightarrow$ as smooth as OPT

Is it competitive?

For any sequence $\sigma$ and $l \leq k - i$:

$$\text{Smoothed-LRU}_{k,i}(\sigma) \leq \frac{k-i}{k-i-l+1} \cdot \text{OPT}_l(\sigma) + l$$

As competitive as LRU for size $k-i$. 
LRU-Random

- Smoothed-LRU and EOA are very smooth, but not competitive

Is there a “reasonable” policy that beats the additive $\delta(k + 1)$ lower bound?

- LRU-Random: evict the $i^{th}$ oldest page with probability $\frac{1}{iH_k}$

LRU-Random is $k$-competitive (against adaptive adversary)

LRU-Random is $\left(1, \left(1 + \frac{11}{6}\right) \delta\right)$-smooth for $k = 2$

$$1 + \frac{11}{6} = 2.83 < 3 = k + 1$$
LRU-Random: Conjectures

LRU-Random is $(1, \Theta(H_k^2) \delta)$-smooth

LRU-Random is $\Theta(H_k^2)$-competitive against an oblivious adversary
The Whole Picture

Smoothness

\((c, \chi)\)

\((k, \gamma_2)\)

\((O(H_k), \gamma_1)\)

\((1, \delta\beta)\)

\((1, \delta(k+1))\)

\((1, \delta H_k)\)

\((1, 2\delta)\)

\(1\) \(H_k\) \(2H_k - 1\) \(k\) \(c\) \(\infty\)

OPT

PARTITION EQUITABLE

MARK

SMOOTH

Rand. strongly-comp.

FWF

LRU-Random (k=2)

Smoothed-LRU*

LRU

RAND. strongly-comp.

Deterministic strongly-competitive

EOA

* With resource augmentation

c-competitive
Open Problems

• (Generalize smoothness proof for LRU-Random)
• Is there a randomized „LRU-sibling“?

• Are there randomized algorithms that are smooth „with high probability“?

• Are there „less pessimistic“ notions of smoothness?
Thank you for your attention!