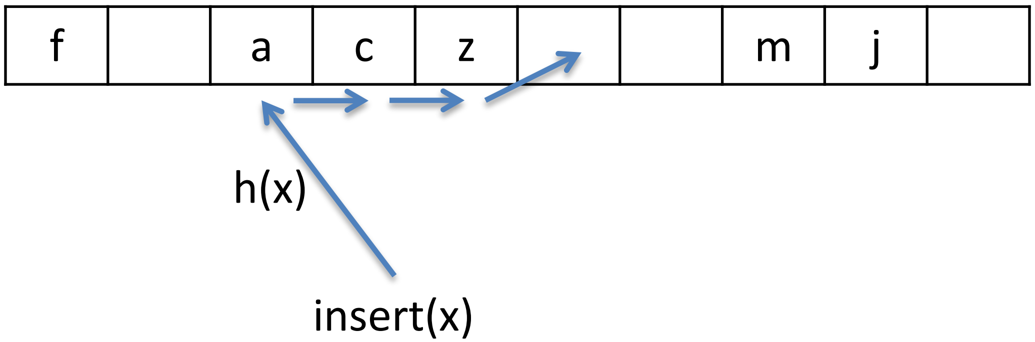
On the *k*-Independence Required by Linear Probing and Minwise Independence

Mihai Pătrașcu   Mikkel Thorup

at&t

ICALP'10

Linear Probing



[Knuth'63] E[time of one operation] = O(1)

"birth of algorithm analysis"

But assumes *h* is a truly random function

⇒ not an algorithm, but a *heuristic*

Implementable Hash Functions

k-independence [Wegman, Carter FOCS'79]

As we draw *h* from a family $\mathcal{H}$:

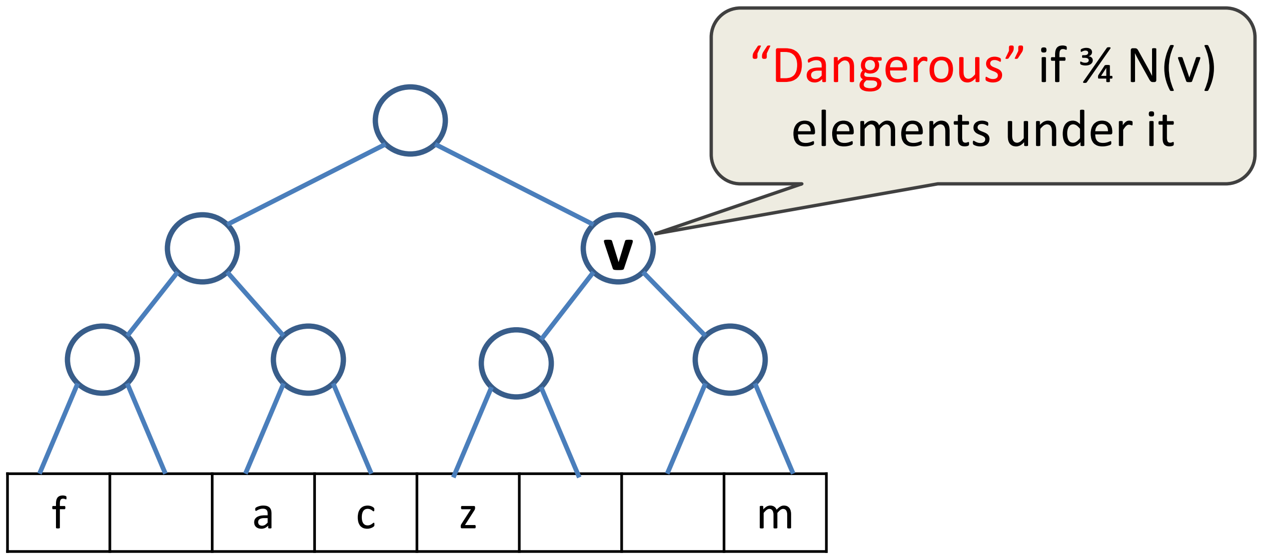
- uniformity: $(\forall)\ x \in U$, $h(x)$ uniform in $[b]$
- independence: $(\forall)\ x_1, \ldots, x_k \in U$, $h(x_1), \ldots, h(x_k)$ i.i.d.

Possible implementation:

- let U = prime field
- draw $a_0, \ldots, a_{k-1} \in U$ randomly
- $h(x) = (\ a_{k-1} x^{k-1} + \ldots + a_1 x + a_0\ ) \bmod b$

Understanding Linear Probing

[Pagh, Pagh, Ružić STOC'07]   5-independence suffices!

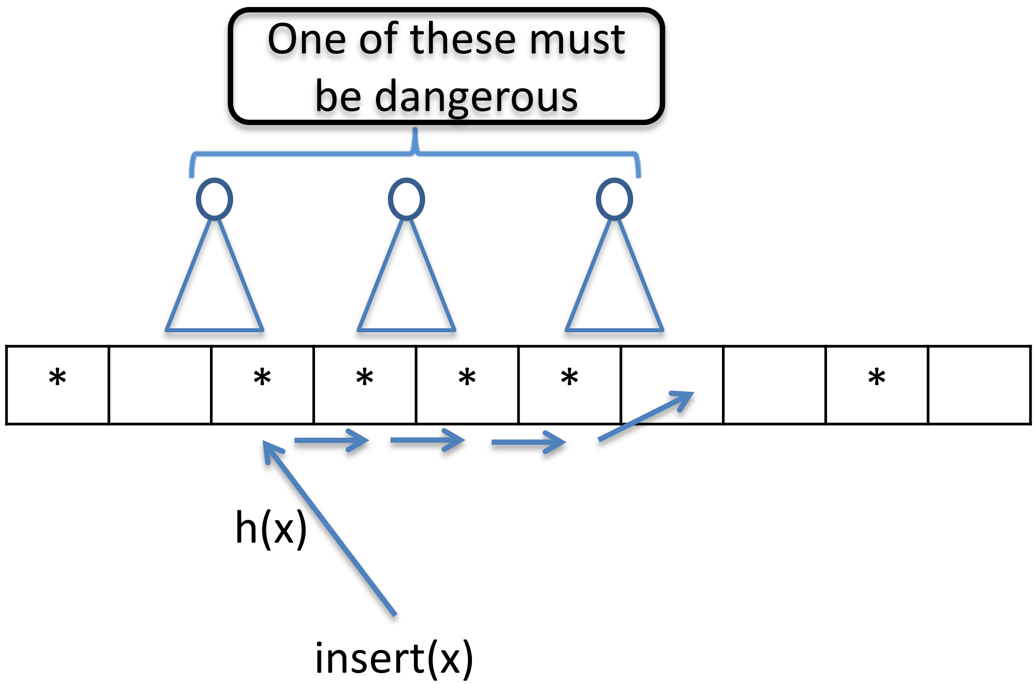


Let N(v) = # leafs under v = $2^{\text{level}(v)}$

Understanding Linear Probing

Main Lemma: If h(x) is in a run of length 2^k

$\Rightarrow$ level k-1 ancestor or a sibling must be dangerous



Understanding Linear Probing

Look at "construction time" = time to insert n elements

Main lemma ⇒ construction time ≤ $\Sigma_{\text{dangerous}}$

Just a classic
balls-in-bins analysis!

E[construction time] ≤ $\Sigma_v \, [N(v)]^2 \cdot \Pr[v \text{ dangerous}]$

2-independence ⇒ Chebyshev bound

⇒ $\Pr[v \text{ dangerous}] \le 1/N(v)$

⇒ E[construction time] ≤ $\Sigma_v \, N(v) = O(n \lg n)$

Understanding Linear Probing

Look at "construction time" = time to insert n elements

Main lemma ⇒ construction time ≤ $\Sigma_{\text{dangerous}}$

Just a classic
balls-in-bins analysis!

E[construction time] ≤ $\Sigma_v\, [N(v)]^2 \cdot \Pr[v \text{ dangerous}]$

4-independence ⇒ 4th moment bound

⇒ $\Pr[v \text{ dangerous}] \leq 1/[N(v)]^2$

⇒ E[construction time] ≤ $\Sigma_v\, O(1) = O(n)$

Understanding Linear Probing

	k=2	k=3	k=4	k=5
Construction time	O(n lg n) Ω(n lg n) [PPR]		O(n)	
Time/operation				

Understanding Linear Probing

	k=2	k=3	k=4	k=5
Construction time	O(n lg n) Ω(n lg n) [PPR]		O(n)	
Time/operation		O(lg n)		O(1)

One query with k-independence

= keys arrange themselves by (k-1)-independence

+ the query hits a random location

Understanding Linear Probing

	k=2	k=3	k=4	k=5
Construction time	$\Theta(n \lg n)$		$\Theta(n)$	
Time/operation	**??**	$O(\lg n)$		$\Theta(1)$

Do we really need "one more" for bounds / operation?

Understanding Linear Probing

	k=2	k=3	k=4	k=5
Construction time	$\Theta(n \lg n)$		$\Theta(n)$	
Time/operation	$\boldsymbol{\Theta(\sqrt{n})}$	$O(\lg n)$		$\Theta(1)$

Do we really need "one more" for bounds / operation?

YES.

Nasty 2-independent family such that:

- often, (∃) run of $\sqrt{n}$ elements;
- the query often falls in this bad run.

Understanding Linear Probing

	k=2	*k*=3	*k*=4	*k*=5
Construction time	$\Theta(n \lg n)$	**??**	$\Theta(n)$	
Time/operation	$\Theta(\sqrt{n})$	$O(\lg n)$		$\Theta(1)$

Could 3-independence help?

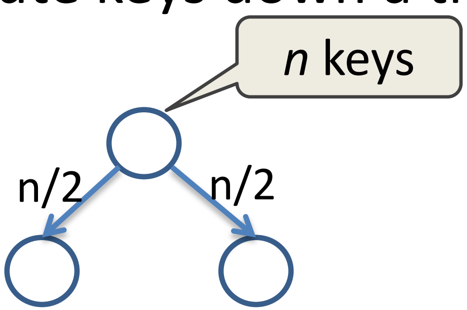
Understanding Linear Probing

	k=2	k=3	k=4	k=5
Construction time	$\Theta(n \lg n)$	**$\Omega(n \lg n)$**	$\Theta(n)$	
Time/operation	$\Theta(\sqrt{n})$	$O(\lg n)$		$\Theta(1)$

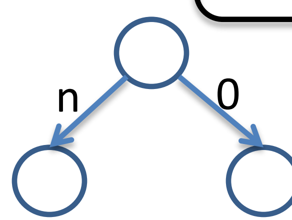
Could 3-independence help? NO

Distribute keys down a tree:

If Pr[Case 2] ≈ 1/n
⇒ 3-independence!



Case 1



Case 2

cost $\Omega(n^2)$

Understanding Linear Probing

	k=2	k=3	k=4	k=5
Construction time	$\Theta(n \lg n)$	$\Theta(n \lg n)$	Θ(n)	
Time/operation	$\Theta(\sqrt{n})$	$\Theta(\lg n)$	**??**	Θ(1)

Can both phenomena hit you simultaneously?

Understanding Linear Probing

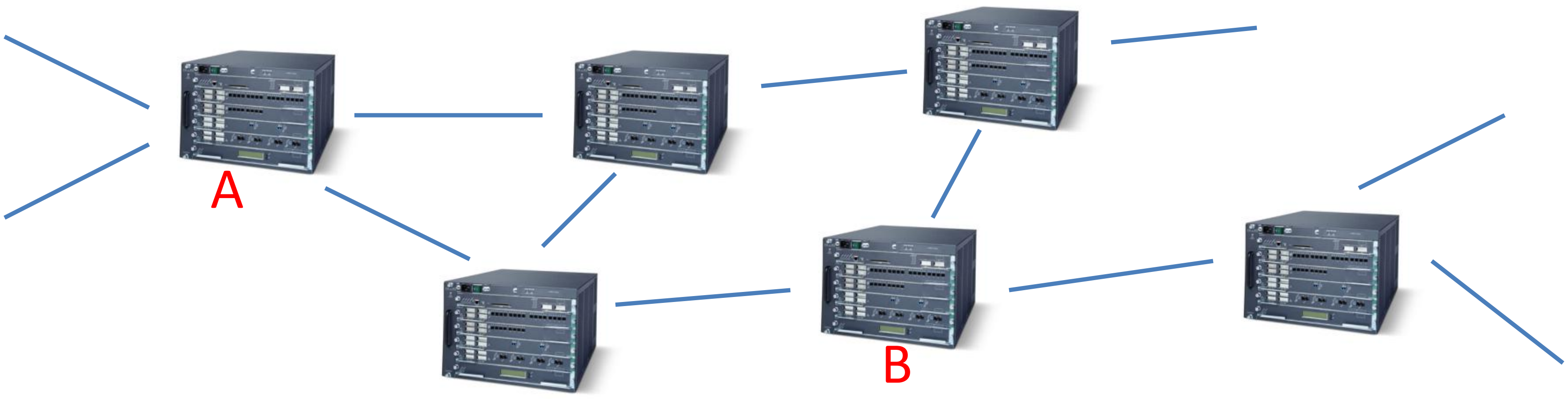
	k=2	k=3	k=4	k=5
Construction time	$\Theta(n \lg n)$	$\Theta(n \lg n)$	$\Theta(n)$	
Time/operation	$\Theta(\sqrt{n})$	$\Theta(\lg n)$	$O(\lg n)$ **$\Omega(\lg n)$**	$\Theta(1)$

Can both phenomena hit you simultaneously? YES

Apply the bad 3-independent distribution *only* on the query path
⇒ 4-independent!

[Nasty proof ☹]

Minwise Independence



Problem: how many packets pass through both A and B?

Jaccard coefficient: $|A \cap B| / |A \cup B|$

Algorithm:

- hash and keep min h(A), min h(B)
- $\Pr[\min h(A) = \min h(B)] = |A \cap B| / |A \cup B|$
- repeat to estimate accurately

Hashing Guarantees

Minwise Independence: for any S, $x \in S$

$$\Pr[\, h(x) = \min h(S)\,] = 1 / |S|$$

Implies: $$\Pr[\min h(A) = \min h(B)] = |A \cap B| \;/\; |A \cup B|$$

☹ Minwise independence not easy to obtain

Hashing Guarantees

ε-Minwise Independence: for any S, $x \in S$

$$\Pr[\, h(x) = \min h(S) \,] = (1 \pm \varepsilon) \,/\, |S|$$

Implies: $\Pr[\min h(A) = \min h(B)] = (1 \pm \varepsilon)\ |A \cap B| \,/\, |A \cup B|$

[Indyk SODA'99] Any $c \cdot \lg(1/\varepsilon)$-independent family is ε-minwise independent

Here: Some $c' \cdot \lg(1/\varepsilon)$-independent families are *not* ε-minwise independent

What it All Means

All our hash families are artificial

… we understand the *k-wise independence concept*

In practice:

- (a * x) >> shift

More results:

- $\Omega(n \lg n)$-construction for linear probing
- terrible minwise behavior

What it All Means

All our hash families are artificial
… we understand the *k-wise independence concept*

In practice:

- (a * x) >> shift
- tabulation-based hashing

Forthcoming paper:
Simple tabulation (3-wise independent) achieves
• linear probing in O(1) time (+ Chernoff concentration!)
• o(1)-minwise independence

What it All Means

All our hash families are artificial

… we understand the *k-wise independence concept*

In practice:

- (a * x) >> shift
- tabulation-based hashing

The polynomial hash function:

- performance not understood
- but not too good in practice…

The End

Open problem: cuckoo hashing

- 6-independence needed [Cohen, Kane]
- O(lg n)-independence suffices