AEC '89

Cairo December 9-12, 1989

A Storage Scheme for an Automated Archive

Lutz M. Wegner
Reda Khalifa

University of Kassel
Department of Mathematics and Computer Science
D-3500 Kassel, West Germany

Outline:

- Infinite Data Files
- Basic Ideas
- Inverted Prefix B-Trees
- Some Implementation Details
- Expected Performance
- Status and Future Plans

Off-line records are identified through smallest identifying prefixes:

Example:

Full keys	Rear Compression	Front Compression
Martha (on-line)		
Martin	$\underrightarrow{\text{Mart}}\text{i}$	4, i
Mary	$\underrightarrow{\text{Mar}}\text{y}$	3, y
Matthew	Mat	2, t
Mavis (on-line)		

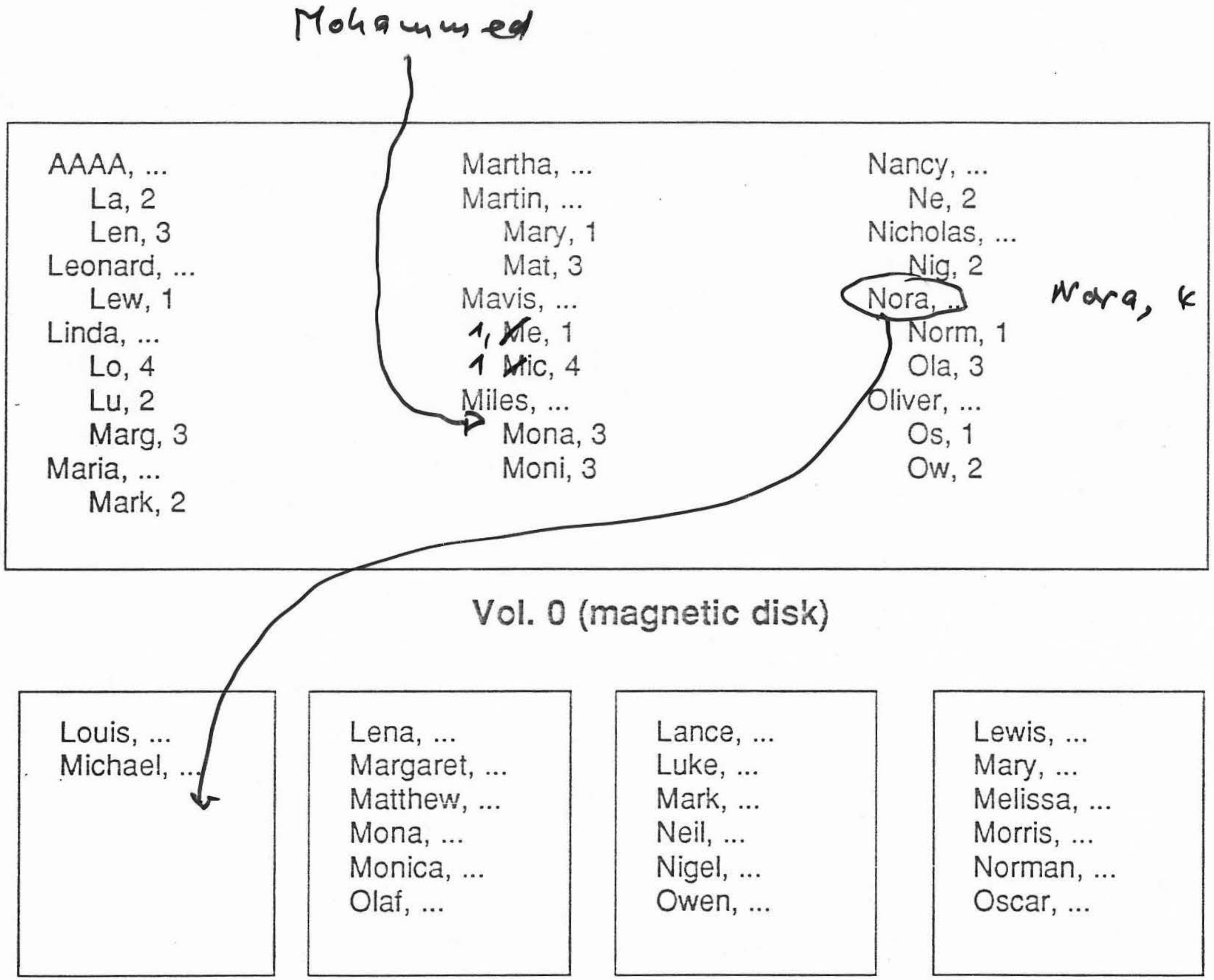
What is the average prefix length?

➡ ln 2 ≈ 1.44 characters [Knuth, vol. 3]

➡ Prefix + 1 Byte Offset + 2 Bytes Vol.No ≤ 5 Bytes

Requires that file on Vol. 0 is in sorted order

➡ Inverted Prefix B-Tree [cf. Bayer/Unterauer 1977]



Mohammed

AAAA, ...
La, 2
Len, 3
Leonard, ...
Lew, 1
Linda, ...
Lo, 4
Lu, 2
Marg, 3
Maria, ...
Mark, 2

Martha, ...
Martin, ...
Mary, 1
Mat, 3
Mavis, ...
1, ~~M~~e, 1
1 ~~M~~ic, 4
Miles, ...
Mona, 3
Moni, 3

Nancy, ...
Ne, 2
Nicholas, ...
Nig, 2
Nora, ...
Norm, 1
Ola, 3
Oliver, ...
Os, 1
Ow, 2

Nora, k

Vol. 0 (magnetic disk)

Louis, ...
Michael, ...

Lena, ...
Margaret, ...
Matthew, ...
Mona, ...
Monica, ...
Olaf, ...

Lance, ...
Luke, ...
Mark, ...
Neil, ...
Nigel, ...
Owen, ...

Lewis, ...
Mary, ...
Melissa, ...
Morris, ...
Norman, ...
Oscar, ...

Vol. 4 (on-line) Vol. 3 (off-line) Vol. 2 (off-line) Vol. 1 (off-line)

Classical Solution

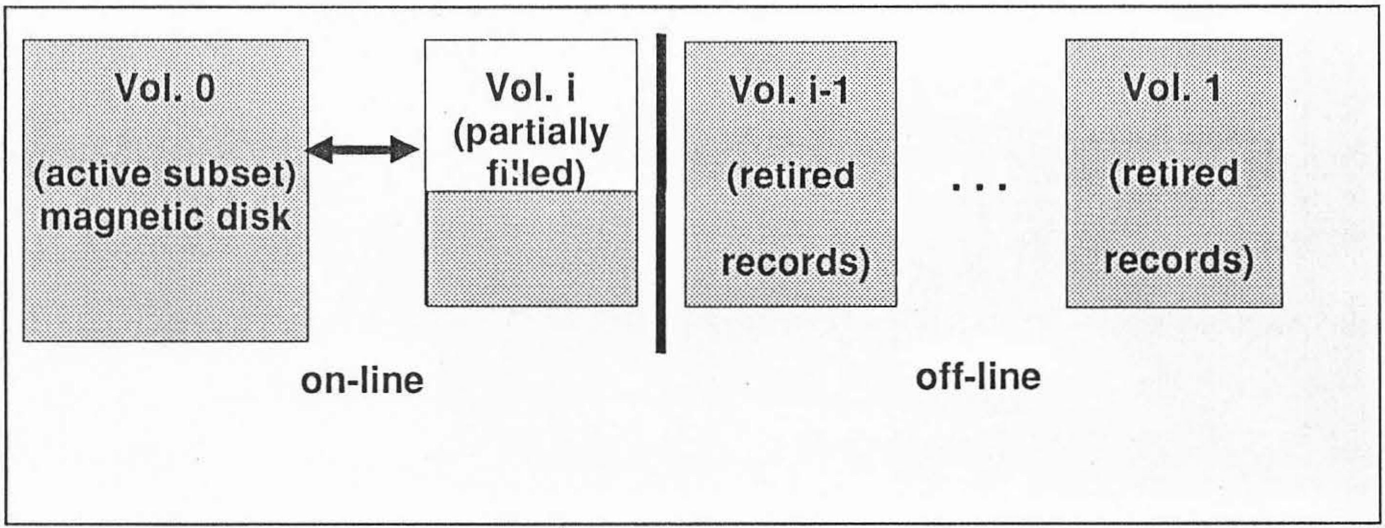
File Reorganization, Time Stamps, Tape Archives:

- time consuming
- error prone
- no direct access
- expensive

➠ not practical

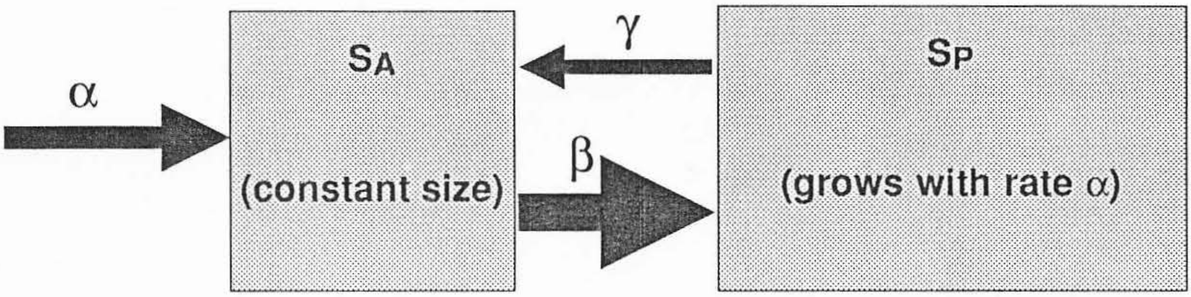
New Solution:

➠ automated migration of passive records (files) into an archive with direct access, possibly on optical disks, using LRU replacement



Problem:

Records or files are added to a file system at a constant rate, but only a constant size subset is *active*. On the other hand, *passive* records cannot be deleted because they cannot be identified or because of legal restraints.

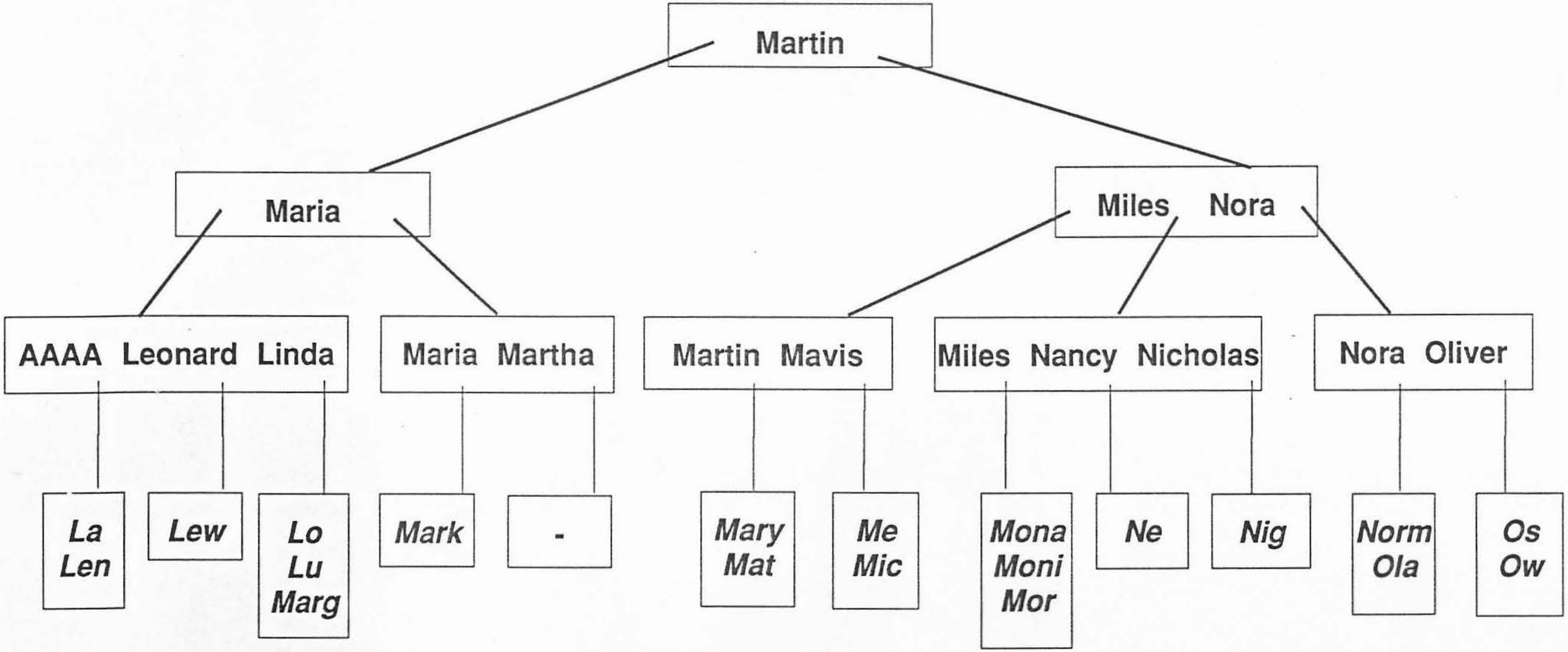


System in steady state: $\beta = \alpha + \gamma$

System practical if $\gamma << \alpha$

Examples:

- Text Processing
- Engineering
- Accounting
- Medical Records
- Program Development


Martin
Maria
Miles Nora
AAAA Leonard Linda
Maria Martha
Martin Mavis
Miles Nancy Nicholas
Nora Oliver
La Len
Lew
Lo Lu Marg
Mark
-
Mary Mat
Me Mic
Mona Moni Mor
Ne
Nig
Norm Ola
Os Ow

Retrieve record r_K with key K

(a) K is key of active record: done

(b) K matches prefix

- (ba) still on-line: look in partially filled volume
- (bb) off-line: fetch volume and insert in drive, move r_K to vol. 0, split prefix list, retire other record, merge prefix list

(c) no match on K: record not in file!

Insert record r_K

(d) K is key of active record: error!

(e) K matches a prefix: fetch off-line volume, check record there, if unequal adjust prefix else error.

(f) no match on K: insert in vol. 0 without adjustments

Delete record r_K

...

Off-line volumes can be write protected or write-once (WORM) optical disks!!

(1) Standard

u	v	w	x	y	z

uv : volume number, uv = 0 implies long volume number
w : offset into key, w = 255 implies long offset
xyz : prefix of length ≤ 3, z = FF implies prefix continued, y or z = NUL implies end of key

(2) Long Prefix, Extra Long Prefix, ...

u	v	w	a_1	a_2	FF

a_3	a_4	a_5	a_6	a_7	a_8

uv, w as in (1)
$a_1 \ldots a_8$ prefix of length ≤ 8, a_8 = FF implies extra long prefix, any of a_4 to a_8 = NUL implies end of key

(3) Long Offset, Extra Long Offset, ..., Long Volume Number, Extra Long Volume Number,

u	v	w	a_1	a_2	a_3

x	y	z_1	z_2	z_3	z_4

uv as in (1), uv = 0 implies long volume number
w as in (1), w = 255 implies long offset
$a_1a_2a_3$ as in (1) and (2)
xy : long offset, xy = 0 implies extra long offset
$z_1z_2z_3z_4$: long volume number, 0 implies extra long volume number
long prefix follows 2nd sextet if necessary

Performance: Example 1

Assume 16 MB hard disk, 2 KB records, 8 Byte prefix on the average, 1 MB floppy (= 512 records per off-line volume)

TABLE II

$\lvert S_A \rvert$	min years on-line	insert. per workday	migration starts after	vol. 0 exhausted after years
4096 records on-line = 1/2 volume = 8 MB	8 y	2	8 y	2048
	4 y	4	4 y	1024
	2 y	8	2 y	512
	1 y	16	1 y	256
	6 m	32	6 m	128
	3 m	64	3 m	64
	6 w	128	6 w	32
6144 records on-line = 3/4 volume = 12 MB	12 y	2	4 y	1024
	6 y	4	2 y	512
	3 y	8	1 y	256
	18 m	16	6 m	128
	9 m	32	3 m	64
	4.5 m	64	6 w	32
	2.25 m	128	3 w	16
7168 records on-line = 7/8 volume = 14 MB	14 y	2	1 y	512
	7 y	4	6 m	256
	3.5 ~~m~~ y	8	3 ~~m~~ [illegible]	128
	21 m	16	6 w	64
	10.5 m	32	3 w	32
	5.25 m	64	1.5 w	16
	2.62 m	128	3.5 days	8

Performance: Example 2 - Medical Practice

Assume 16 MB hard disk, 1 MB off-line volumes, 2 KB records, 4 new patients per workday ($\alpha = 1024$), 20.000 visits per year, Zipf distribution for visits ($p_i = c/i$ with $c = 1/H_n$)

➠ How offen does the receptionist have to reach into the shelf?

- at most 7 times per workday after 10 years
- at most 13 times per workday after 20 years
- at most 15 times per workday after 30 years

80 visits/ workday

always including the 4 accesses for the insertions!

TABLE III

years after start-up	file size	off-line accesses per year due to retrieval	 insertion	Σ p$_i$ ($i \leq 7168$)
0 - 8	8192	0	0	1
9	9216	550.6	1024	0.972
10	10240	772.5	1024	0.961
11	11264	968.9	1024	0.952
20	20480	2115.0	1024	0.894
30	30720	2816.8	1024	0.859
512	524288	6518.5	1024	0.674

Status

System is operational as stand-alone application
optical disks not yet included

Plans

Provide application interface for C-programs under UNIX
Provide retirement per file (not per record) basis

Experience

System is designed to run "forever" - have you ever written software routines which you know for sure will not be invoked long after you will be dead?