

From Several Hundred Million to Some Billion Words: Scaling up a Corpus Indexer and a Search Engine with MapReduce

Jordi Porta

Computational Linguistics Area

Departamento de Tecnología y Sistemas

Centro de Estudios de la Real Academia Española

(porta@rae.es)

Outline

**“The more observations we gather about language use,
the more accurate description we have about language itself”**

Lin & Dyer, 2010: *Data-intensive text processing with MapReduce*

- A Case Study: *Elpais.com*
- Basic Data Structures for Complex Queries
- Introduction to MapReduce
- The Scale out vs. Scale up Dilemma
- Scaling up with Phoenix++
- Experiments introducing MapReduce into a CMS:
 - Query Engine:
 - Word Counting
 - Word Cooccurrences
 - N-grams
 - Indexer
 - File Inversion
 - Text Inversion
 - Structure Inversion

**“The only thing
better than big data
is bigger data”**



A Case Study: *Elpais.com* on-line archive

EL PAÍS
ARCHIVO EDICIÓN IMPRESA Hemeroteca

VIERNES, 16 de abril de 2010

La erupción de un volcán islandés paraliza los cielos de Europa

La ceniza obliga a cancelar miles de vuelos en una docena de países

WALTER OPPENHEIMER 16 ABR 2010

Archivado en: Retrasos transporte CANCELACIONES transporte Islandia Erupciones volcanes Incidencias transporte Escandinavia Europa Desastres naturales Desastres Sucesos Transporte

La nube de ceniza procedente de la erupción de un volcán en Islandia dejó ayer en tierra a decenas de miles de personas en al menos una docena de países europeos, entre ellos España. Más de 5.000 vuelos fueron cancelados por el cierre de aeropuertos del norte de Europa. Las autoridades islandesas no descartan que el caos se prolongue hasta el domingo. La nube, de entre 6 y 11 kilómetros de longitud, ha forzado a Reino Unido e Irlanda a cerrar sus espacios aéreos, algo inédito en la historia de ambos países. Ni siquiera sucedió tras los atentados del 11-S, que llevó a Gobiernos como el británico a suspender todos los vuelos como medida de precaución.

El volcán en erupción, situado bajo el glaciar Eyjafjalla, ha lanzado nubes de ceniza a una altura de entre 5.500 y 11.000 metros. Las nubes no representan un problema para la salud porque la ceniza llegará al suelo de forma diseminada. Pero sí son una amenaza mayúscula para la aviación porque las partículas de roca, cristal y arena pueden dañar el fuselaje y, sobre todo, bloquear los reactores. Fue eso lo que ocurrió en 1982 cuando un Boeing 747 de British Airways que sobrevolaba un volcán en Indonesia sufrió durante varios minutos el bloqueo de sus cuatro reactores, aunque los pilotos los volvieron a encender y pudieron aterrizar sin problemas. Desde entonces, no se vuela a través de las cenizas.

EL PAÍS KIOSKO MÁS ACCESO A SUSCRIPTORES » Accede a EL PAÍS y todos sus suplementos en formato PDF enriquecido

SECCIONES: PRIMERA INTERNACIONAL ESPAÑA ECONOMÍA OPINIÓN VIÑETAS SOCIEDAD CULTURA TENDENCIAS GENTE OBITUARIOS DEPORTES ÚLTIMA

EDICIONES: ANDALUCÍA CATALUÑA C. VALENCIANA GALICIA MADRID PAÍS VASCO

SUPLEMENTOS: CINE EP3



Nube de ceniza formada tras la erupción de un volcán bajo el glaciar Eyjafjalla, en Islandia. / REUTERS

Base Corpus:

- *Elpais.com* on-line archive:
- 1976–2011
- 2.3 million news
- Cleaned + PoS tagged
- 23 million structural elements
- 878 million tokens



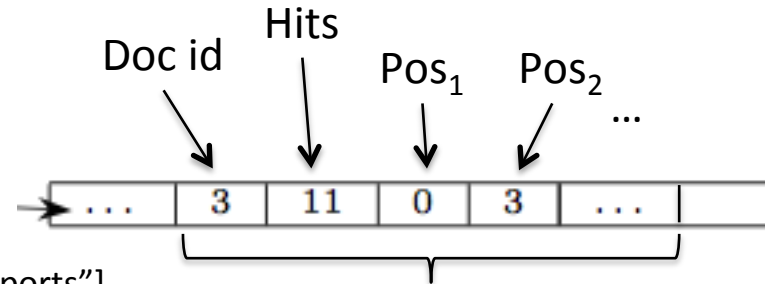
- Lists of linguistic elements:
 - Concordances
 - Counts
 - Associations
 - Distributions
 - ...
- Sub-corpora
- Non-trivial processing:
 - Cooccurrence networks / clustering / ...
 - Word vectors
 - ...

Basic Data Structures for Corpus Indexing Supporting Complex Queries

- Textual indices:

- Posting Lists:

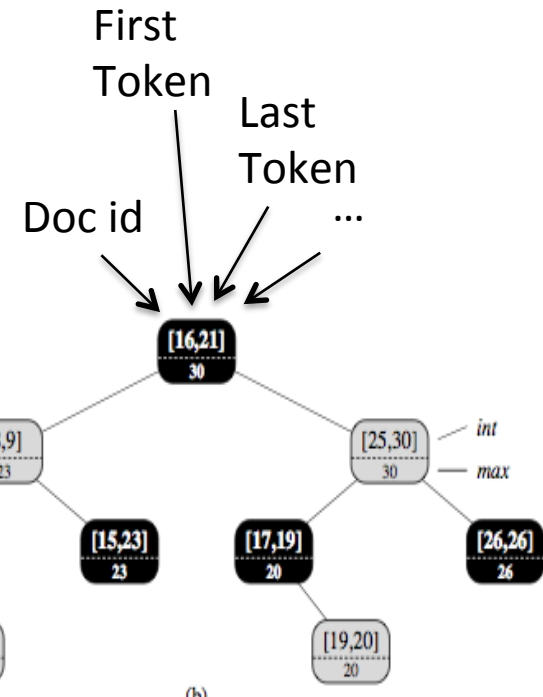
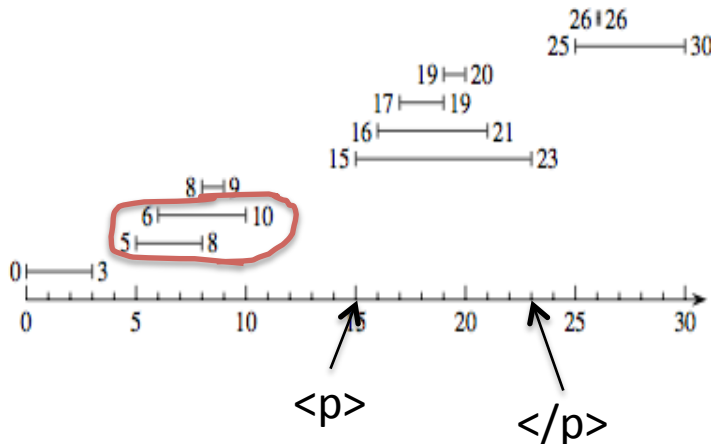
- [token="de"]
 - [token="la" and msd="Nfs"]
 - [token="pelota"] within [section="sports"]
 - ...



- Structural indices:

- Interval Trees:

- P within TEXT
 - P containing HI[REND=IT] containing [lemma="de"]
 - ...



How Text is Indexed

Documents:

- begin,
- end,
- offset
- ...

-	-	0	1	2	3	4	5	6	7	8	9	10	...
<text>	<p>	La	nube	de	cenizas	,	procedente	de	la	erupción	de	un	...
<text>	<p>	la	nube	de	ceniza	,	procedente	de	la	erupción	de	un	...
<text>	<p>	Tfs	Nfs	P	Nfp	,	Afs	P	Tfs	Nfs	P	Qms	...

docs	words	lemmas	msds
0 ... 51	... 100 101 102 103 104 105 106 107 108 109 110 ...	... 3 10 19 32 43 45 19 3 60 19 69 ...	... 6 15 22 41 43 56 22 6 15 22 72 ...
1 ... 75	... 0 10 19 24 43 45 19 3 60 19 69 ...	... 3 10 19 32 43 45 19 3 60 19 69 ...	... 6 15 22 41 43 56 22 6 15 22 72 ...
2 ... 91	... 0 10 19 24 43 45 19 3 60 19 69 ...	... 3 10 19 32 43 45 19 3 60 19 69 ...	... 6 15 22 41 43 56 22 6 15 22 72 ...
3 ... 100	... 0 10 19 24 43 45 19 3 60 19 69 ...	... 3 10 19 32 43 45 19 3 60 19 69 ...	... 6 15 22 41 43 56 22 6 15 22 72 ...
...	...	...	...

135M

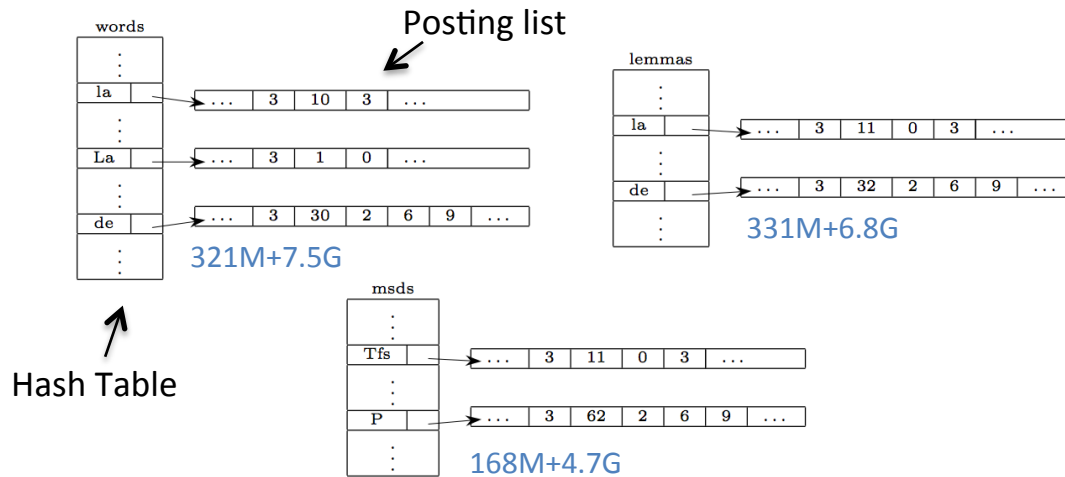
3.7G

54M

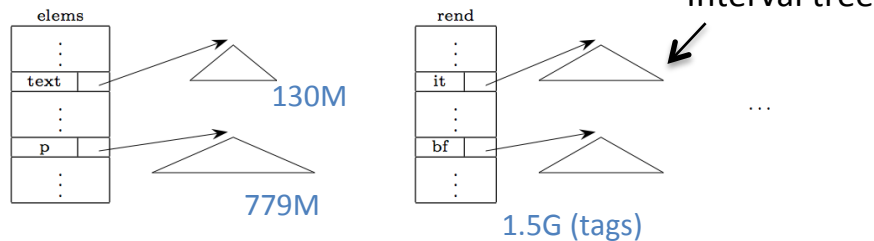
Corpus Layers

String Singletons

Positional Indices



Structural Indices



What is MapReduce?

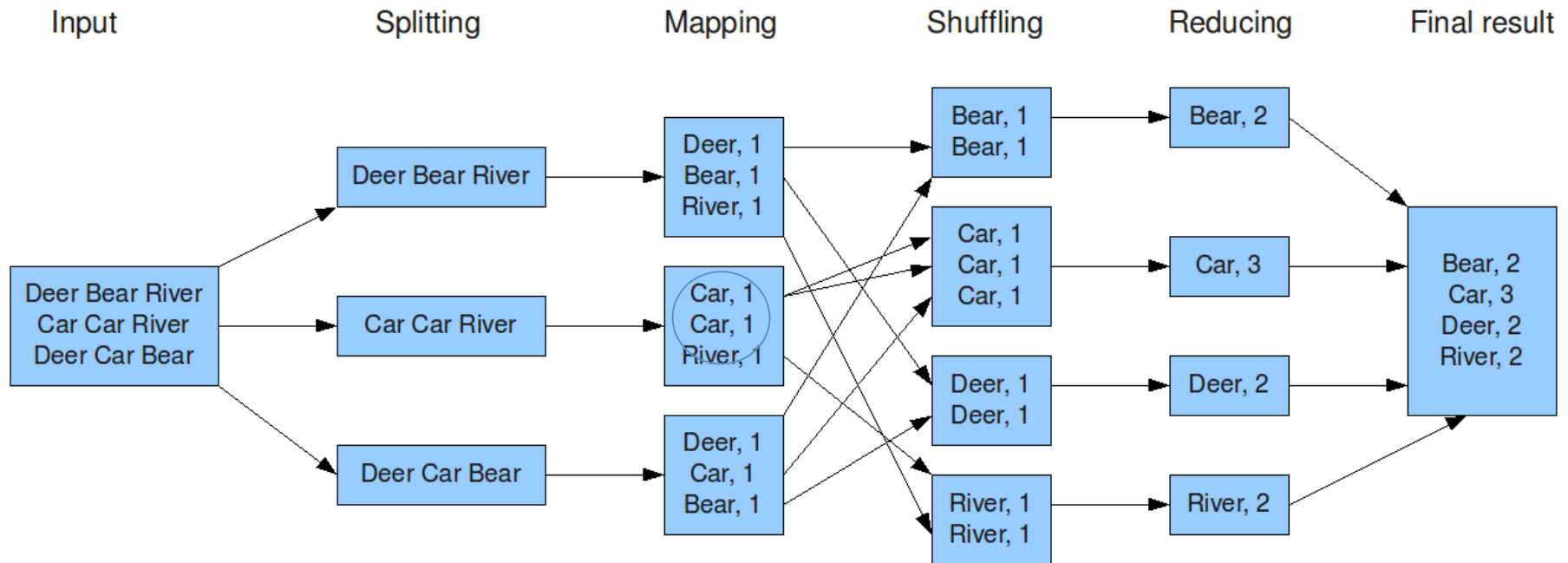
- A widely used parallel computing model
- For batch processing of data on terabyte or petabyte scales
- On clusters of commodity servers (typically)
- Fault tolerance: task monitoring and replication
- Abstracts parallelization, synchronization and communication
- Hadoop is the most popular MapReduce scale out implementation

Hadoop MapReduce

Map(k_1, v_1) $\rightarrow$ list(k_2, v_2)

Reduce($k_2, \text{list}(v_2)$) $\rightarrow$ list(v_2)

The overall MapReduce word count process



Scaling out vs Scaling up Dilemma

- Scaling out (horiz.)
 - Cluster of commodity servers
 - Pros:
 - Cheaper
 - Fault tolerance
 - Easy to update
 - Cons:
 - Power consumption
 - Cooling
 - Space
 - Difficult to implement
 - Networking equipment
 - Licensing costs
- Scaling up (vert.)
 - Powerful server (+proc./+RAM)
 - Pros:
 - Power consumption
 - Cooling
 - Space
 - Easy to implement
 - Networking equip. ~ 0
 - Licensing costs
 - Cons:
 - Expensive
 - Fault tolerance
 - Difficult to update

Scaling out vs Scaling up Dilemma

- Appuswamy & al. (2014) “Scale-up vs Scale-out for **Hadoop**: Time to rethink?” In *Proc. 4th Ann. Symp. on Cloud Computing*.

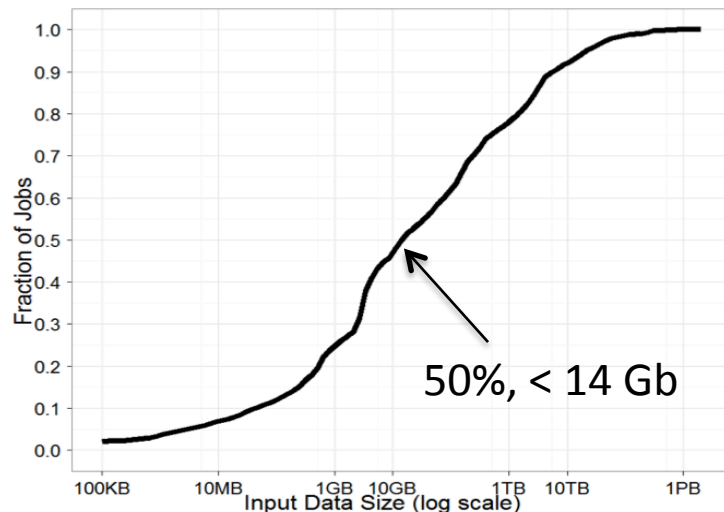


Figure 1: Distribution of input job sizes for a large analytics cluster

	Workstation ¹	Server ¹
Base spec	E5520	4x E7-4820
- CPU	4 cores, HT, 2.3 GHz	32 cores, HT, 2.0 GHz
- Memory	12 GB	0 GB
- Disk	1 HDD (160 GB)	2 HDD (600 GB)
Base cost	\$2130	\$17380
SSD cost	\$270 ²	\$2160 ²
DRAM cost	none	\$5440 ^{2,3} (512GB)
Total cost	\$2400	\$24980
Power	154 W	877 W ⁴

¹ Prices from Dell's website, <http://www.dell.com/>

² Prices from <http://www.newegg.com>

³ 16 GB 240-Pin DDR3 1333 SDRAM.

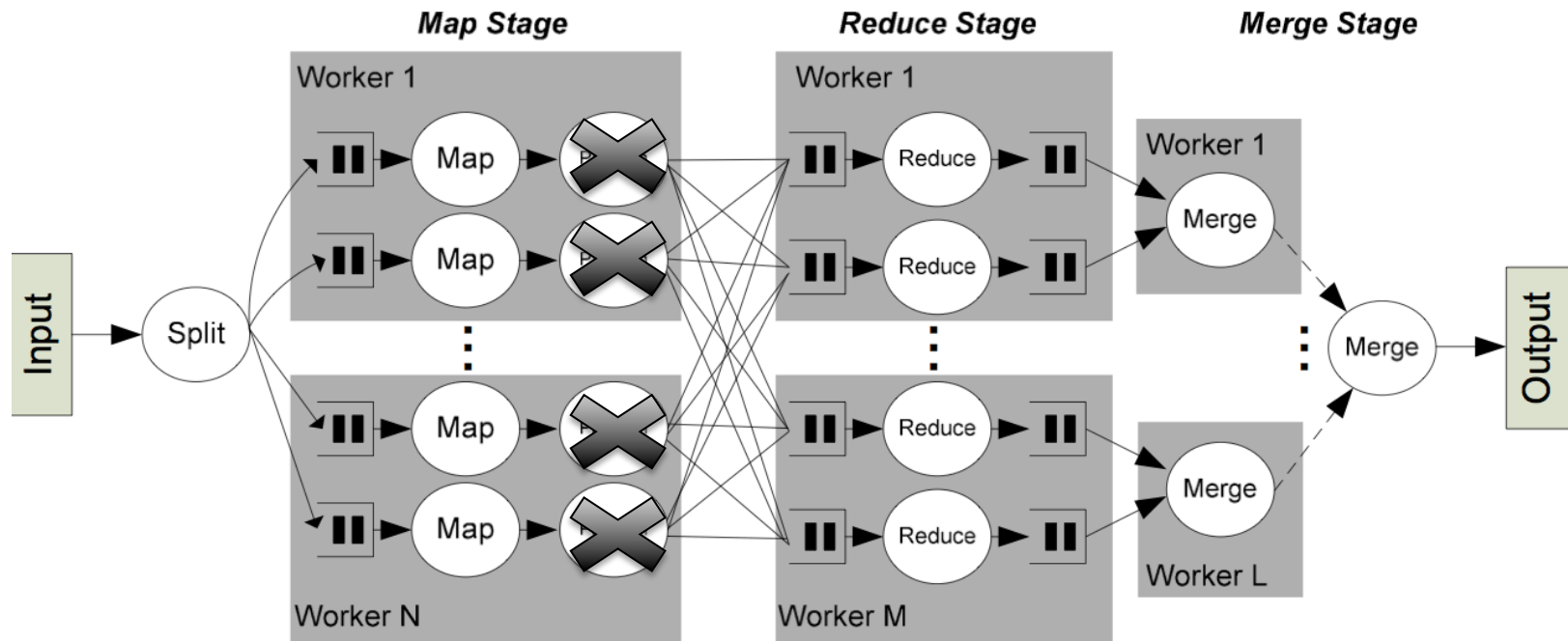
⁴ Measured with 512GB of RAM and all (HDD and SSD) drives connected.

* All processors are Intel Xeon

Table 3: Comparison of workstation (used for scale-out) and server (scale-up) configurations. All prices converted to US\$ as of 8 August 2012.

- Input size in analytics production clusters:
 - Microsoft / Yahoo: **median < 14 Gb**, Facebook: **90% < 100 Gb**
- Conclusion:
 - “Scaling up is more cost-effective for many real-world problems”**

Scaling up with **Phoenix++** MapReduce



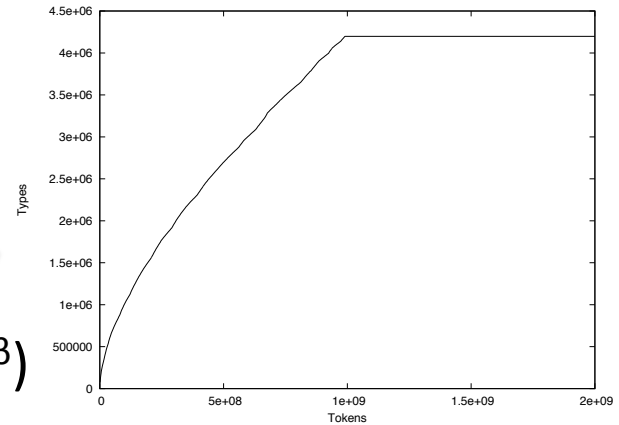
"Phoenix++: Modular MapReduce for Shared-Memory Systems", 2011: Justin Talbot, Richard M. Yoo, and Christos Kozyrakis. *Second International Workshop on MapReduce and its Applications*.

(Code: <http://mapreduce.stanford.edu>)

- Split: divides input data to chunks for map tasks
- **Combiner**: are thread-local objects
- Map: invokes the Combiner for each key-value pair emitted (not at the end of Map)
- ~~Partition/Suffle~~: not in Phoenix++
- Reduce: aggregates key-value pairs in Combiners (All Map task are finished)
- Merge: sort key-value pairs (optional phase)
- **Problems**: introduces rehash between map and reduce phases

Experiments: Data & Machinery

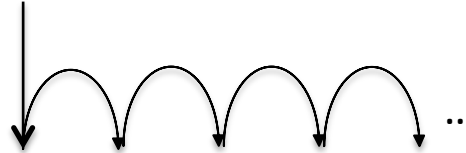
- Base Corpus
 - *Elpais.com* on-line archive (1976–2011)
 - 2.3 million news
 - Cleaned + PoS tagged
 - 23 million structural elements
 - 878 million tokens
- Multiples of *Elpais.com*
 - x2, x4, x8, x16
 - Don't obey Heaps' Law ($V(n) = K \cdot n^\beta$)
- 1x Dell PowerEdge R615
 - RAM: 256 Gb
 - CPUs: 2 x Xeon E5-2620 / 2 GHz (24 threads)
 - DISKS: 3 x SAS 7.200 rpm, RAID-5



Word Counting

Given a (sub-)corpus,
compute the frequencies
of all the words
(word frequency profile)

Word Counting



	docs	
0	...	51
1	...	75
2	...	91
3	...	100
⋮	⋮	⋮

	...	100	101	102	103	104	105	106	107	108	109	110	...
words:	...	0	10	19	24	43	45	19	3	60	19	69	...
lemmas:	...	3	10	19	32	43	45	19	3	60	19	69	...
msds:	...	6	15	22	41	43	56	22	6	15	22	72	...

strings: 0 3 6 10 15 19 22 24 32 39 43 45 56 60 69 72
 La\l\la\l\fs\nube\Nfs\de\P\cenizas\ceniza\Nfp\,\,\procedente\Afs\erupción\un\Qms\...

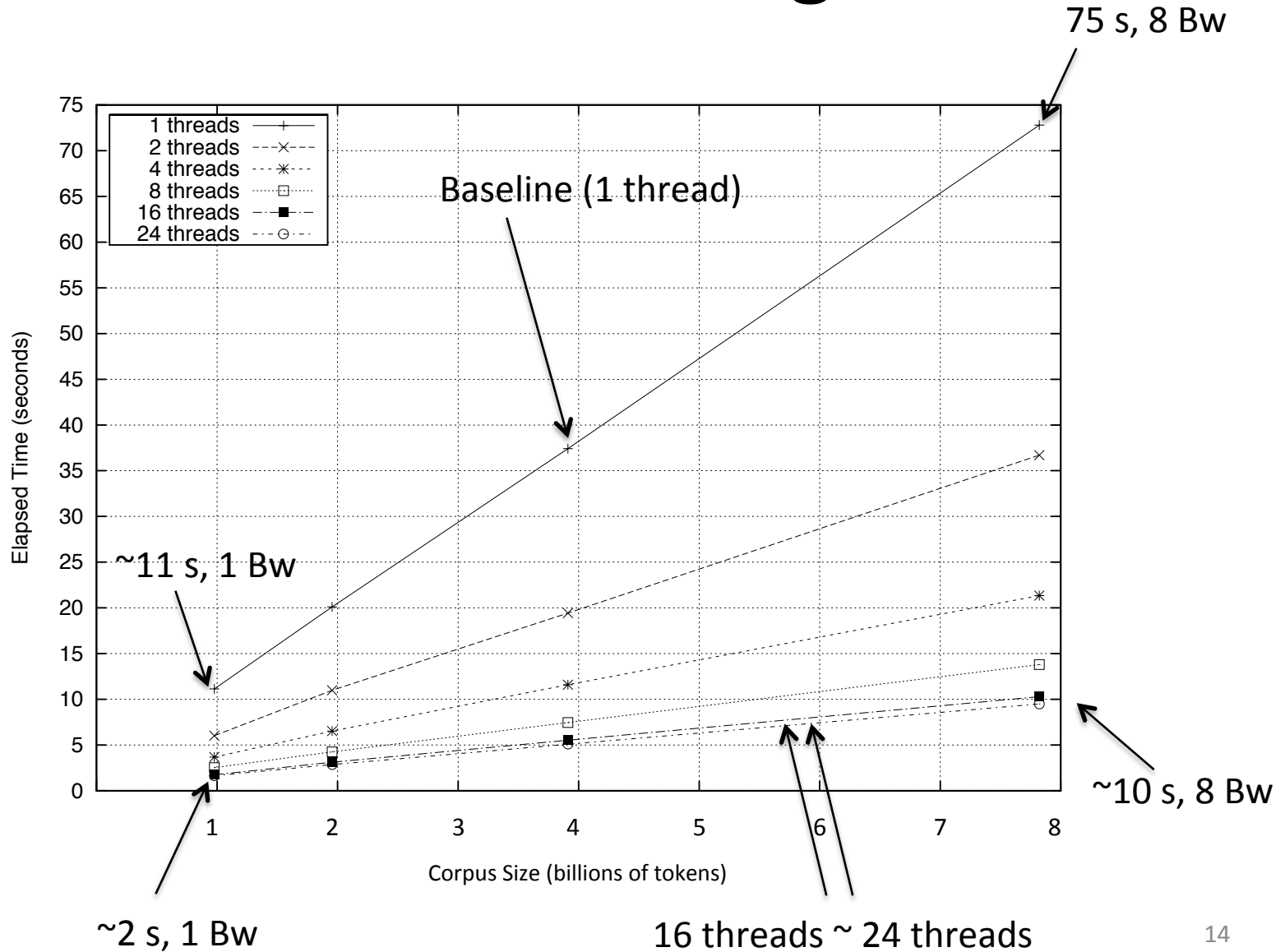
Split(docs) =
 divide docs into <doc, begin, end>

Map(<doc, begin, end>) =
 for i = begin to end do
 emit(words[docs[doc].offset + i], 1)
end

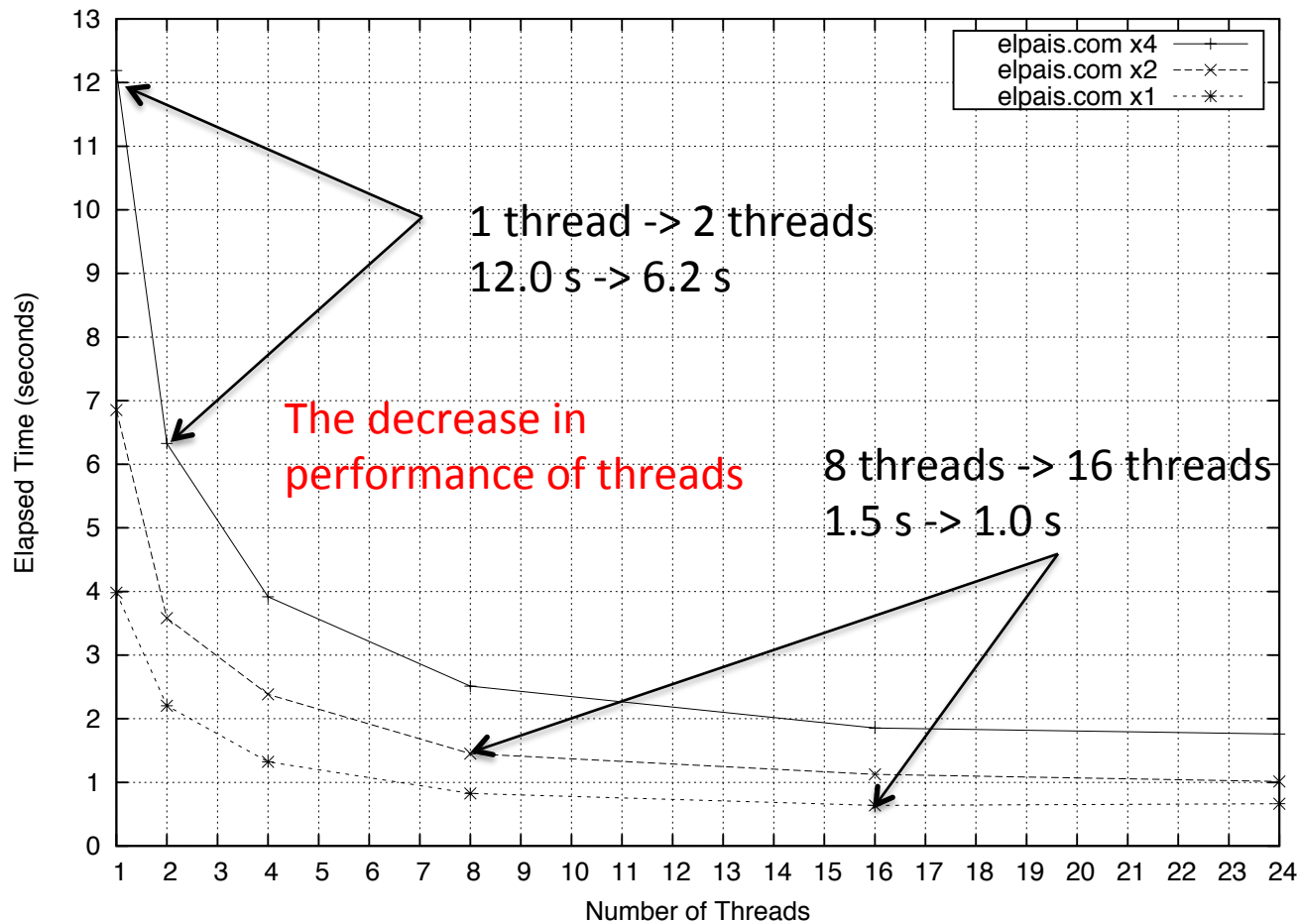
Combiner:

Word	Freq.
0 (la)	456
19 (de)	5667
...	...

Word Counting



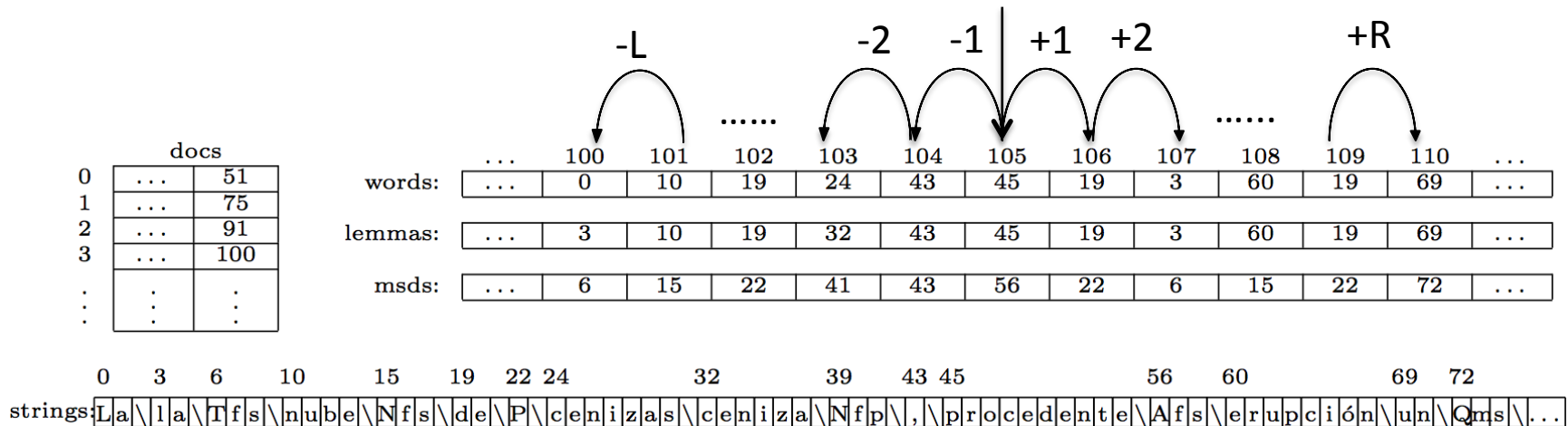
Word Counting



Word Cooccurrences

Given a word (or a lema or a PoS, ...),
compute the frequency of the
surrounding words (left/right)

Word Cooccurrences



Split(Posting list) =
divide the posting list into <doc, n, positions>

Combiner:

Word	Freq.
0 (la)	456
19 (de)	5667
...	...

Map(<doc, n, pos>) =

```

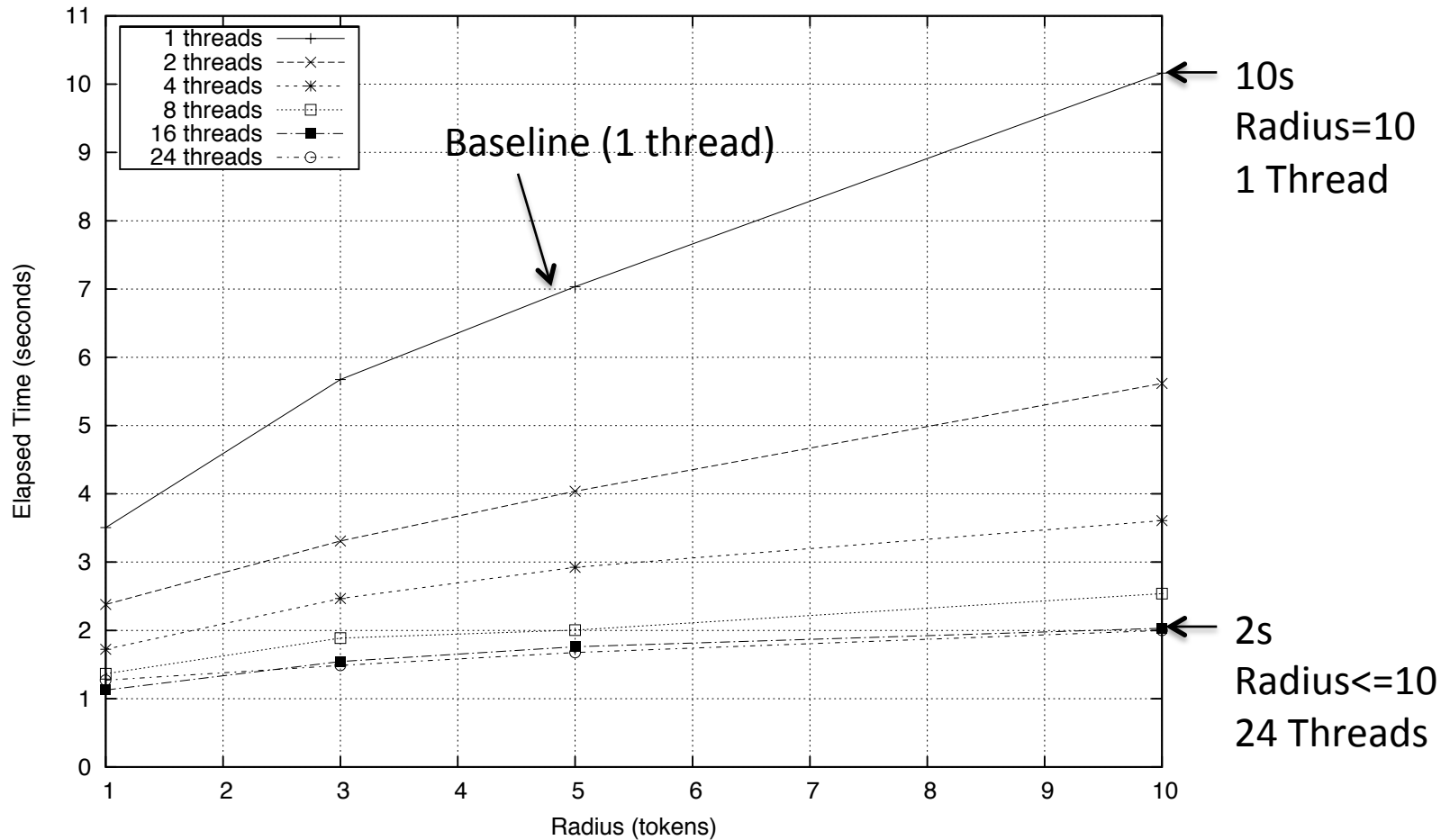
for i = 0 to n - 1 do
  for j = 1 to L do
    emit(words[docs[doc].offset + pos[i] - j], 1)
  end
  for j = 1 to R do
    emit(words[docs[doc].offset + pos[i] + j], 1)
  end
end
  
```

Left Context (for j = 1 to L)

Right Context (for j = 1 to R)

Word Cooccurrences

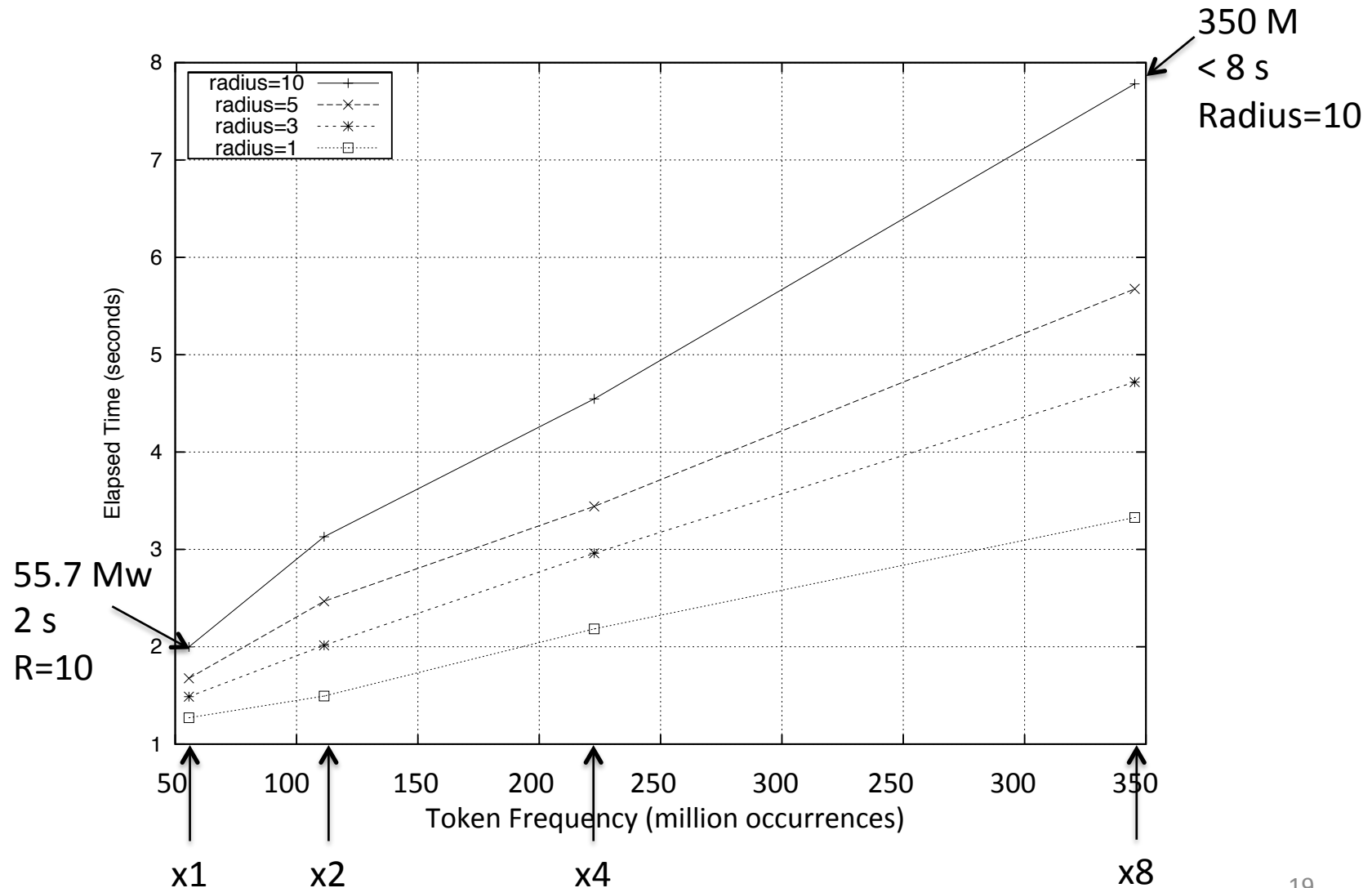
of [token = "de"] in Elpais.com x1 => 55.7 Millions



$L = R = \text{Radius}$

Word Cooccurrences

of [token = "de"] in Elpais.com xN using 24 threads



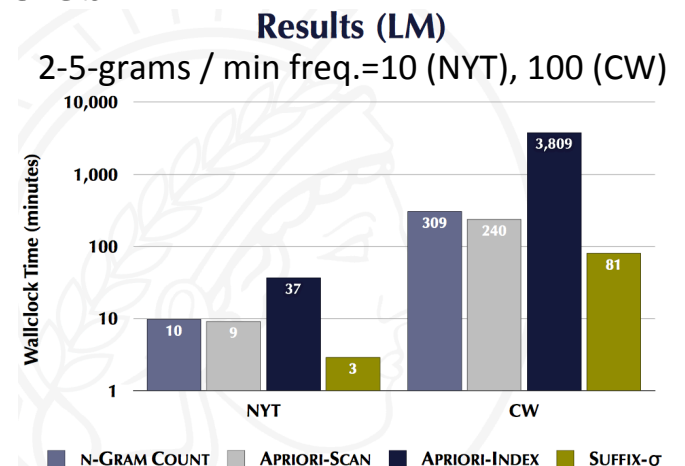
N-grams

Given a (sub-)corpus,
compute the frequency of all the sequences of
words (lemma, PoS, ...) with lengths between a
minimum and a maximum

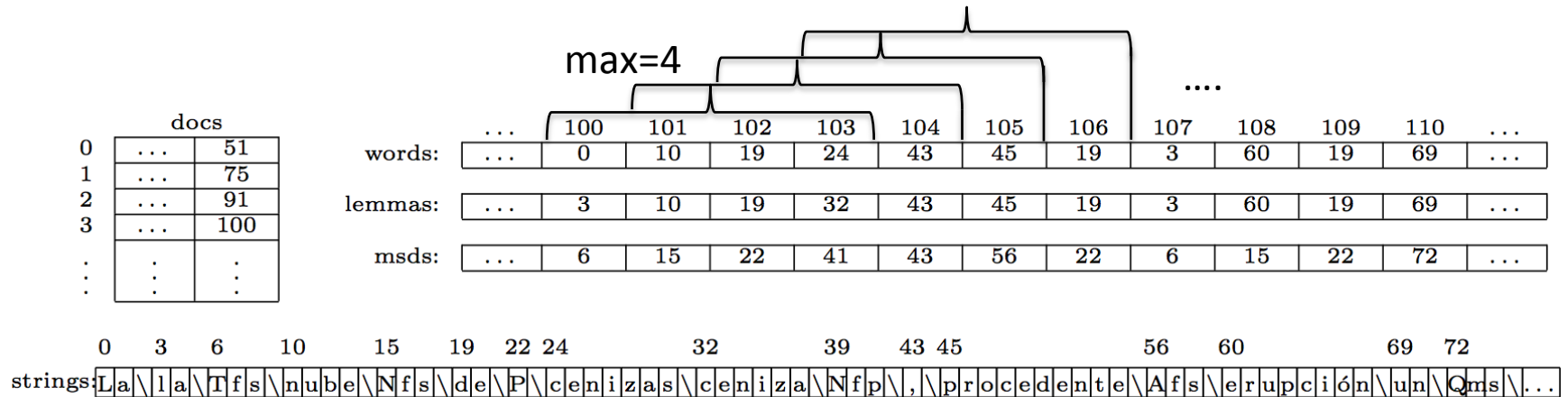
MapReduce N-grams

The Suffix- σ Algorithm

- Klaus Berberich and Srikanta Bedathur (2013): “Computing n-Gram Statistics in MapReduce”. In *Proceedings of 16th International Conference on Extending Database Technology (EDBT 2013)*
 - Stage 1: Maximal n-gram counts are computed and sorted
 - Stage 2: Shorter n-gram counts are computed from the suffix array of maximal n-grams in the reducer tasks
- New York Times Annotated Corpus (NYT)
 - 1.8 million newspaper articles, 1987-2007, ~3 Gb
 - **1 billion tokens** (345,827 types)
- ClueWeb09-B (CW)
 - 50 million web documents, 2009, ~246 Gb
 - **~21 billion tokens** (979,935 types)
- **Hadoop Cluster: 10 nodes**
 - **2x6 cores, 64 Gb RAM, 4x2 Tb HDD**



MapReduce N-grams



Split(Interval tree) =
divide the interval tree into intervals:
<doc, begin, end>

Combiner:

Word	Freq.
<0,10,19,24> (la,nube,de,cenizas)	456
<19,24,43,45> (de,cenizas,"",",procedente)	5667
...	...

Map(<doc, begin, end>) =
for pos = begin to end do

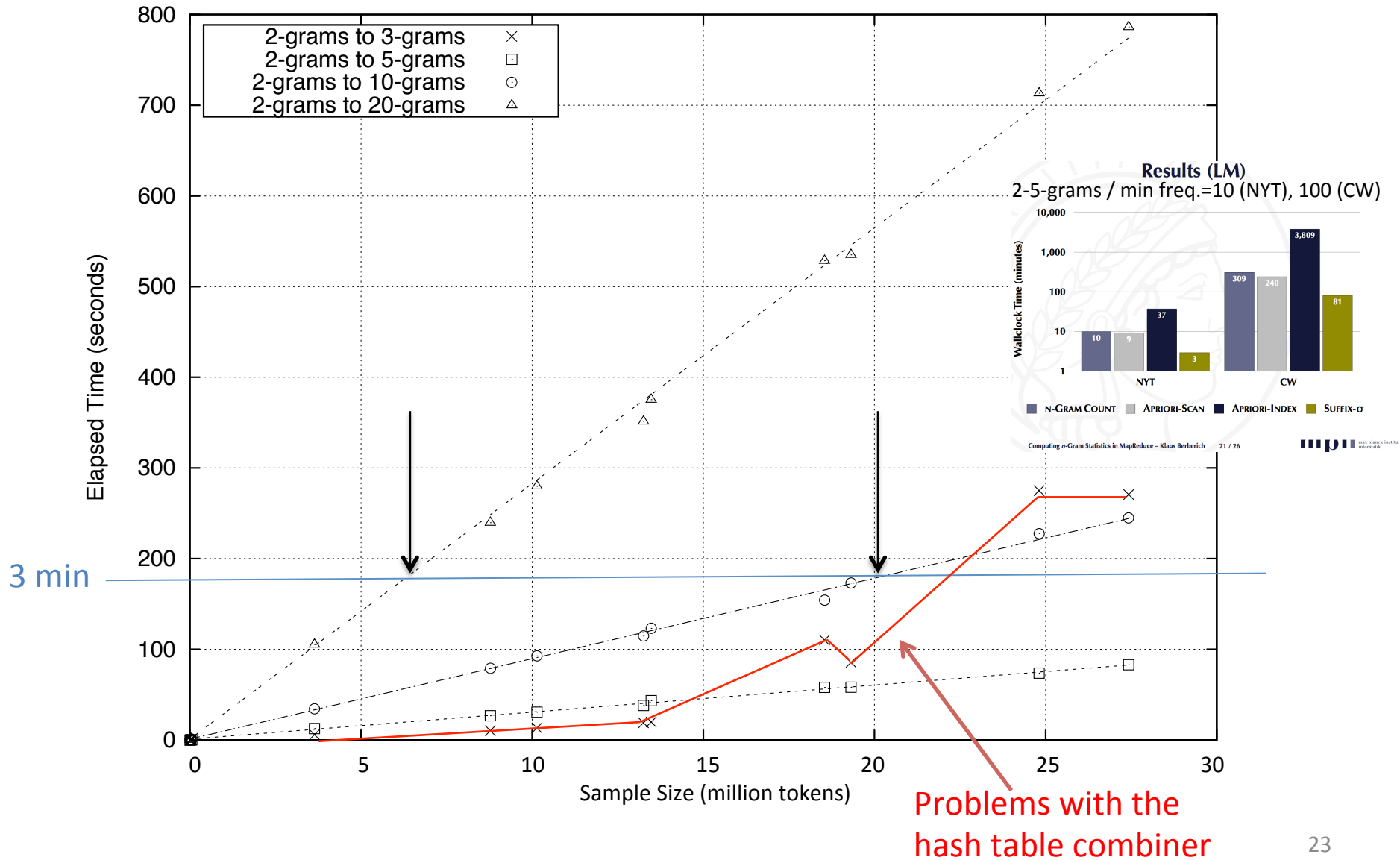
emit(<words[docs[doc].offset + pos,
words[docs[doc].offset + pos + 1,
...
words[docs[doc].offset + pos + max>],

1)

end

Maximal
n-gram

MapReduce N-grams

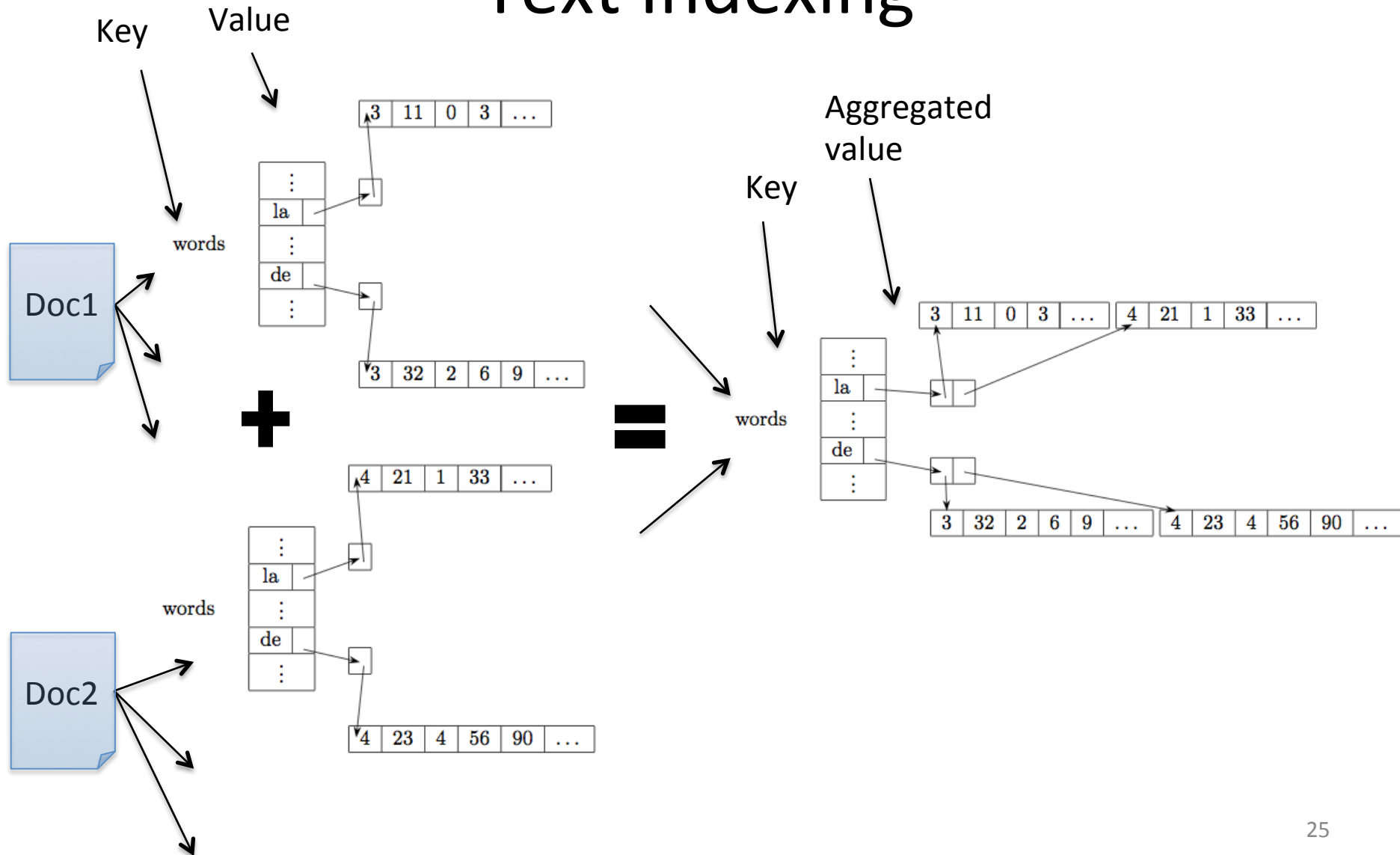


MapReduce into the Indexer: File Inversion

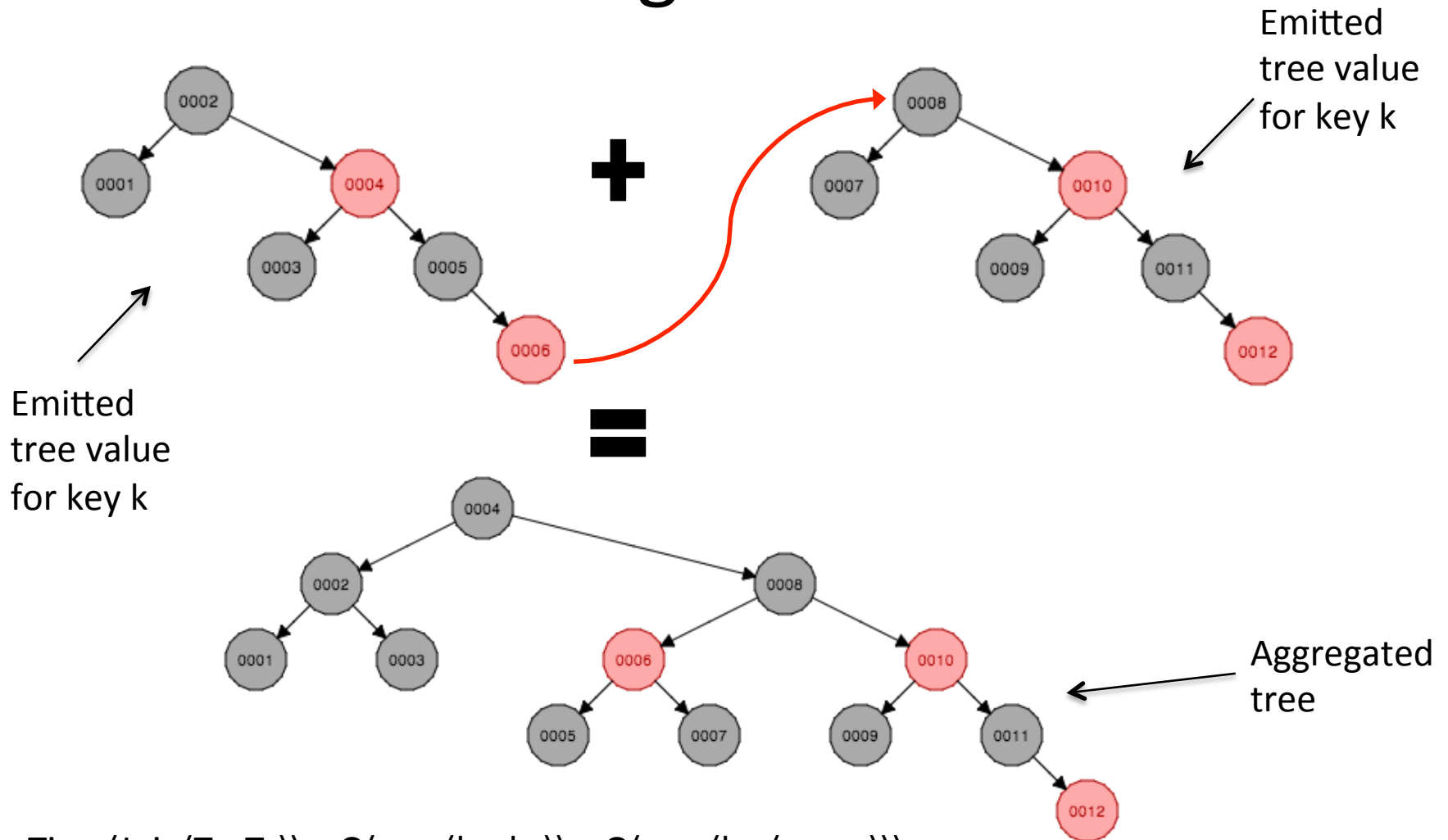
- Text inversion
- Structural inversion

MapReduce File Inversion

Text Indexing

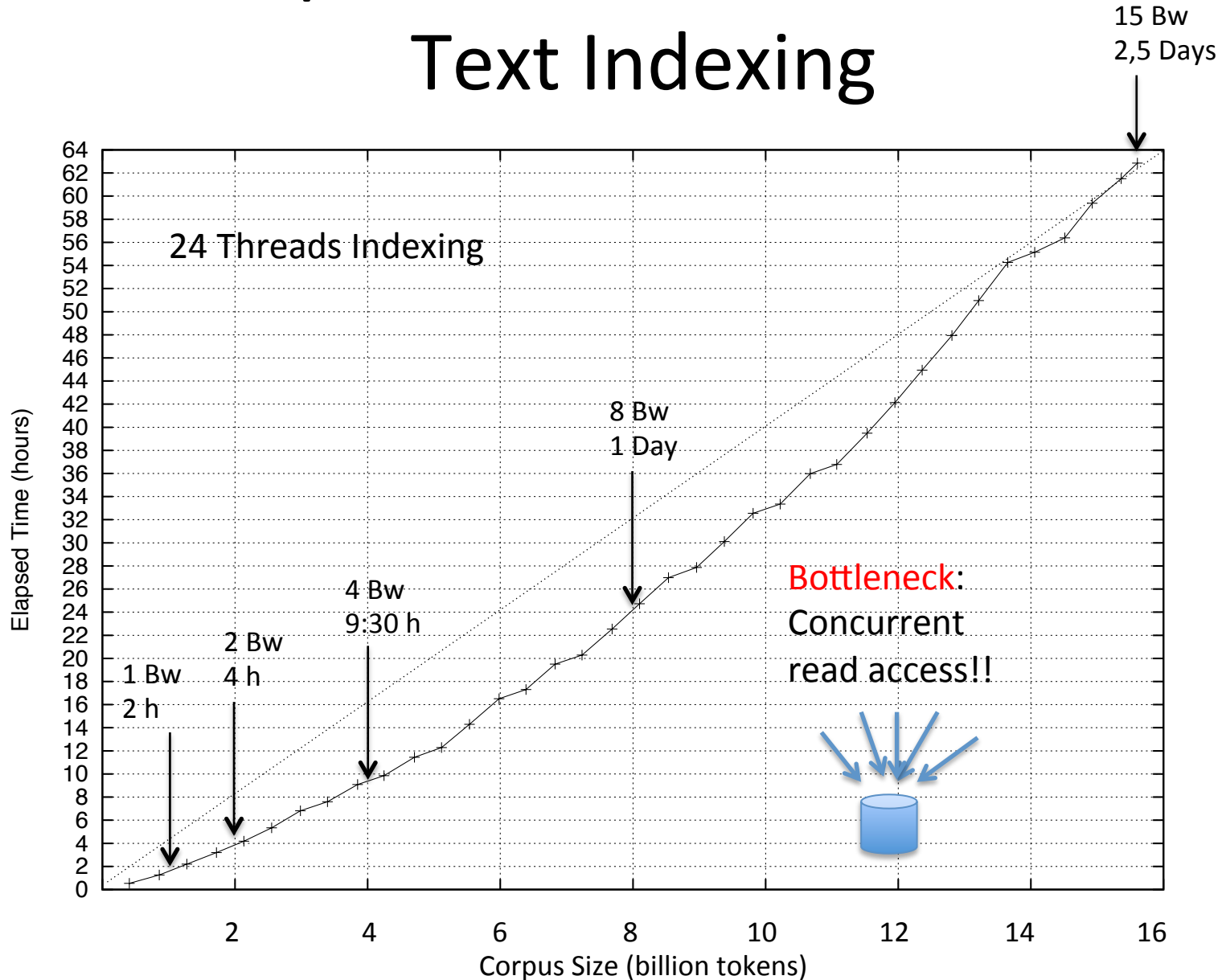


MapReduce File Inversion: Indexing Structure



$$\text{Time}(\text{Join}(T_1, T_2)) = O(\max(h_1, h_2)) = O(\max(\log(n_1, n_2)))$$

MapReduce File Inversion: Text Indexing



Conclusions & Future Work

- Conclusions:
 - Applications scale well to the sizes of interest using MapReduce
 - MapReduce in multicore servers is a cost-effective solution for corpus management tools
- Future Work:
 - Improve hash containers
 - Introduce new associative containers
 - Implement other non-trivial statistics

Thank you !!

Questions ??