600 JAHRE
UNIVERSITÄT LEIPZIG

Matching Anatomy Ontologies

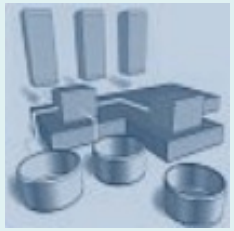
- Quality and Performance

Anika Groß

IZBI UNIVERSITÄT LEIPZIG

Interdisciplinary Centre for Bioinformatics
http://www.izbi.uni-leipzig.de

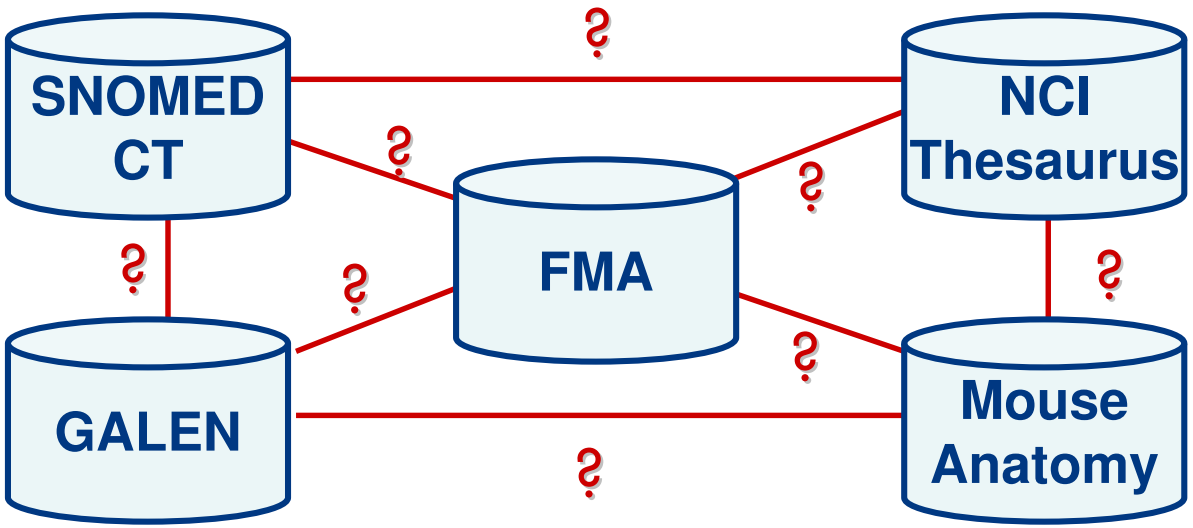
Database Group Leipzig
http://dbs.uni-leipzig.de



Leipzig, 9th December 2009

Ontologies

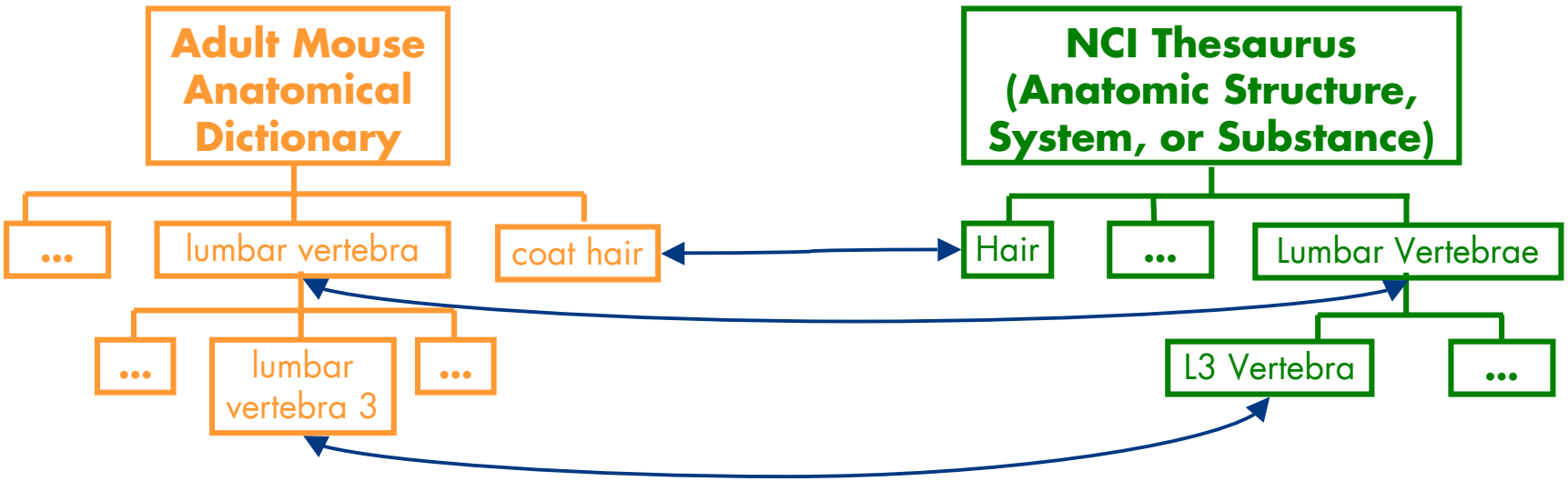
- Formal representation to model the entities and their relationships within a domain
- Life science example domains
  - Biological processes
  - Anatomy
  - ...
- Often multiple ontologies for the same domain, e.g.: Open Biomedical Ontologies provide 34 anatomy (related) ontologies



→ **Overlapping information?**
→ **Need for creating mappings**

Ontology Mappings

- Set of semantic correspondences between concepts of different ontologies
- Automatically generated by ontology matching approaches
- Crucial for
  - Data integration
  - Enhanced data analysis across ontologies
  - …



Overview

- Ontologies, ontology mappings
- Matching of anatomy ontologies
  - Dataset
  - Match workflow (stepwise)
  - Evaluation w.r.t. quality (precision, recall, …)
- Distributed Matching
  - Resources
  - Distributed Architecture
  - Evaluation w.r.t. performance
- Conclusion & future work

Matching Anatomy Ontologies

Aim Find overlapping information in anatomy ontologies

Possible application Aligning anatomies across model organisms to compare functional information about genes/proteins

Example Alignment (OAEI Anatomy track)

- Mouse anatomy: Adult Mouse Anatomical Dictionary (MA)
- Human anatomy: Anatomy subset of NCI Thesaurus (NCI)

Perfect mapping contains 1523 correspondences

Given partial reference mapping

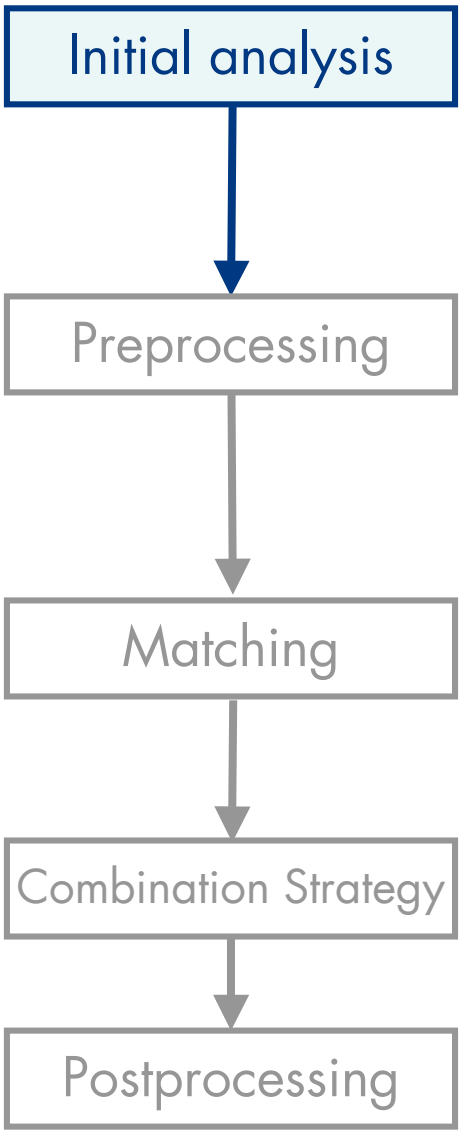
- contains 988 correspondences (934 trivial, 54 non-trivial)
- "Real" evaluation of my match results:

*"[…] it is possible to send us an submission […] you will be informed about precision, recall and f-measure […] we restrict ourselves to do this for each system only two times […]"**

- **Match result should be as good as possible before I try …**

* http://oaei.ontologymatching.org/2007/results/anatomy/samples.html

Workflow



- Basic statistics of input ontologies
- Frequency of word occurrence
- Origin of words (language), e.g. English, Latin, Greek
- Depth of graph structure
- …

Initial Analysis

- **Basic statistics**

	MA	NCI
# concepts	2737	3298
# relationships	3438	5423
# synonyms (per attribute)	344 (0.1)	5247 (1.6)
# attr = #names + #synonyms	3081	8545

- **Frequent word analysis (Top 10)**
- occ_{word} – # occurrences of word in attribute set (names, synonyms)

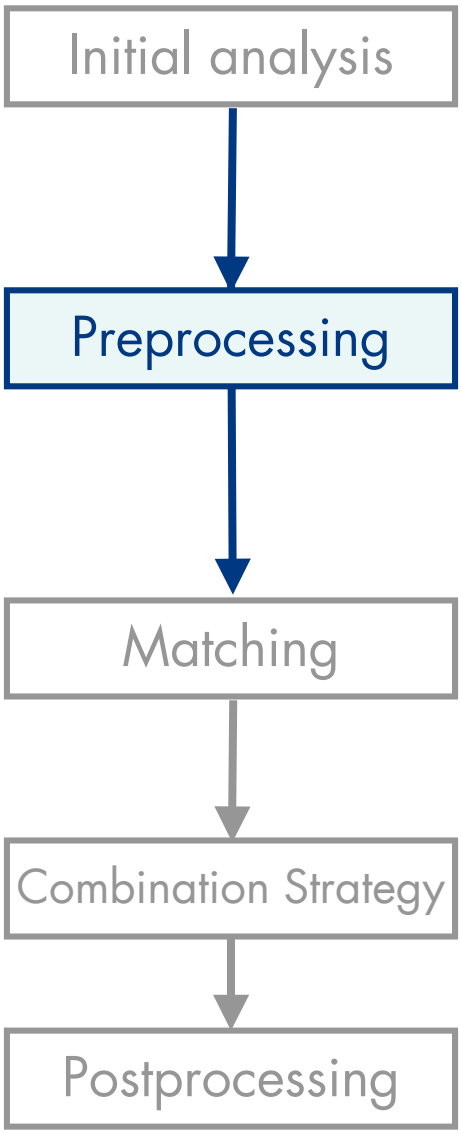
Words in MA	occ	occ /|attr|	Words in NCI	occ	occ /|attr|
vein	180	0,058	of	1079	0,126
artery	168	0,055	the	722	0,084
muscle	163	0,053	cell	718	0,084
nucleus	122	0,040	artery	504	0,059
bone	113	0,037	vein	269	0,031
nerve	111	0,036	muscle	263	0,031
gland	104	0,034	system	261	0,031
system	100	0,032	tissue	218	0,026
epithelium	92	0,030	gland	205	0,024
of	79	0,026	bone	199	0,023

→ semi-automatic elimination of most frequent terms that carry no valuable information (e.g. „of")?

TODO

- Application of this knowledge in preprocessing
- Analyze origin of words (language), e.g. English, Latin, Greek
  → used to find further synonyms
- Depth of graph structure …

Workflow



- Basic statistics of ontologies
- Frequency of word occurrence
- Origin of words (language), e.g. English, Latin, Greek
- Depth of graph structure

Attribute normalization

- Remove delimiters / punctuation
- Upper case to lower case
- Roman to arabic numbers …

Precompute n-Level name path, …

Get additional knowledge n Google-URLs, UMLS Metathesaurus, …

Preprocessing

Attribute Normalization

- Remove all delimiters/punctuations, replace upper case letters
  Right_Lung_Respiratory_Bronchiole → right lung respiratory bronchiole

TODO → Conversion to simplest word part (stemming)
→ Alphabetic order

Precomputation of attribute values

- n-Level name path: path from concept to root including n levels
  → Consider all non-redundant paths (is_a, part_of) to the root

Example Get 3-Level name path for concepts „NCI_C12719“, „MA_0001161“

ID: NCI_C12719
Name: Ganglion*
Path: /nervous system/peripheral nervous system/ganglion

ID: MA_0001161
Name: peripheral nervous system ganglion
Path: /nervous system/ganglion/peripheral nervous system ganglion

* **Ganglion** German: Ein von einer Kapsel umschlossener Nervenzellknoten;
Spanish: Formaciones nodulares que hay en el trayecto de los nervios

Preprocessing

Precomputation of attribute values

- Get additional knowledge n Google-URLs (different query strategies)
- E.g. compare only wiki-URLs

Query "pallidum" site:en.wikipedia.org

ID: MA_0000889
Name: pallidum
1st-Wiki-URL: http://en.wikipedia.org/wiki/Globus_pallidus

Query "globus pallidus" site:en.wikipedia.org

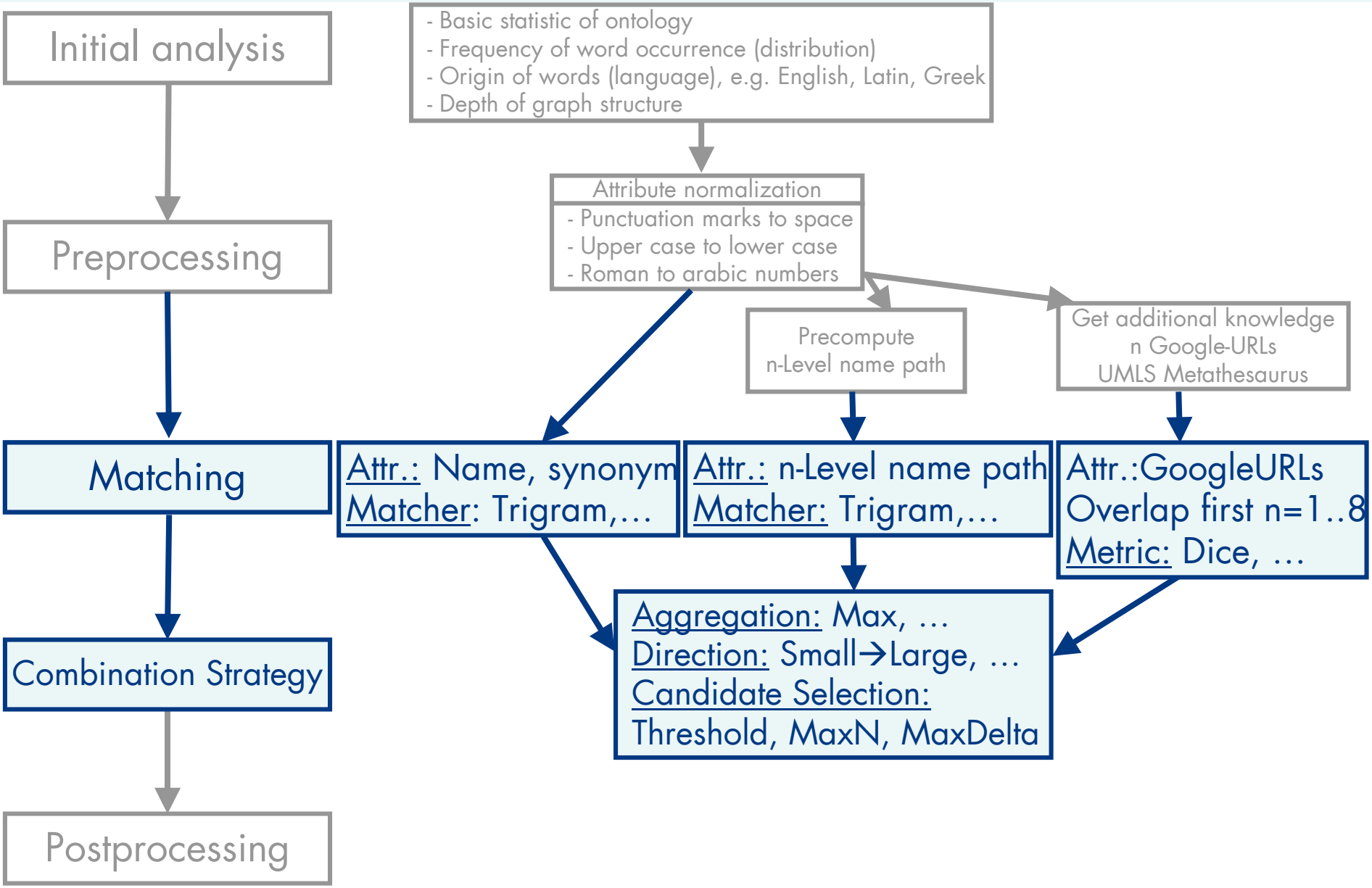
ID: NCI_C12449
Name: Globus_Pallidus *
1st-Wiki-URL: http://en.wikipedia.org/wiki/Globus_pallidus

TODO

- Try different query strategies, e.g.
  (anatomy OR anatomical structure) „globus pallidus"
- Use UMLS-Metathesaurus
  = comprehensive thesaurus of biomedical concepts

* **Globus pallidus** German: Teil des Mittelhirns; Spanish: parte del cerebro medio

Workflow


Initial analysis
- Basic statistic of ontology
- Frequency of word occurrence (distribution)
- Origin of words (language), e.g. English, Latin, Greek
- Depth of graph structure
Preprocessing
Attribute normalization
- Punctuation marks to space
- Upper case to lower case
- Roman to arabic numbers
Precompute
n-Level name path
Get additional knowledge
n Google-URLs
UMLS Metathesaurus
Matching
Attr.: Name, synonym
Matcher: Trigram,…
Attr.: n-Level name path
Matcher: Trigram,…
Attr.:GoogleURLs
Overlap first n=1..8
Metric: Dice, …
Aggregation: Max, …
Direction: Small→Large, …
Candidate Selection:
Threshold, MaxN, MaxDelta
Combination Strategy
Postprocessing

Matching

- Look for 1,523 correspondences

→ Mapping size should be only slightly more/less

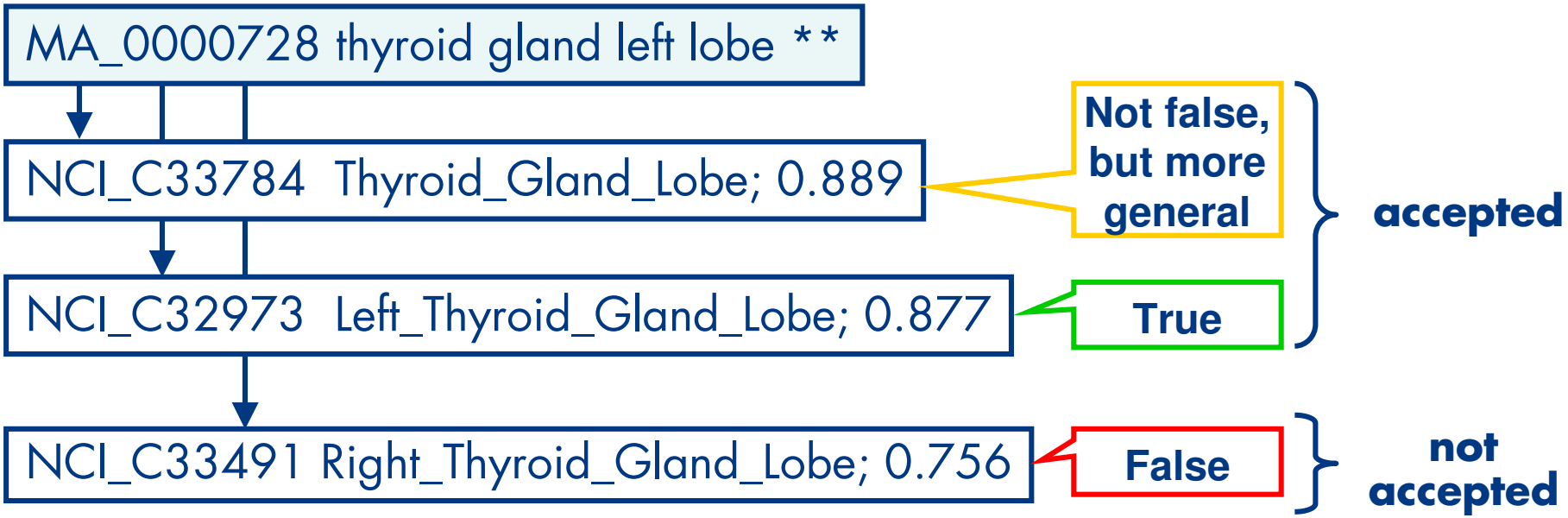
- Search predominantly 1:1 correspondences for anatomy concepts

- Use normalized and precomputed attributes to match MA and NCI
  - name&synonym
  - n-Level name path
  - n googleURLs (only wikipedia)
- Similarity computed using
  - Trigram
  - Overlap (dice)
- Combination
  - Aggregation: union of single matcher results (max similarity)
  - Candidate Selection: Max1, MaxDelta

MaxDelta-Threshold

- MaxDelta strategy*: only correspondences with maximal (and slightly differing) similarity (e.g., 0.03)
- From small to large ontology 

- Example: Correspondences for one MA-concept (used matcher: 2-level name path TRIGRAM, minimal threshold 0.75)



* Do, H.H.; Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches,Proc. VLDB, 2002

** **thyroid gland left lobe** German: linker Schilddrüsenflügel; Spanish: lóbulo tiroideo izquierdo

Evaluation Setup

Combination of matchers:

- TRIGRAM: Name&Synonym, n-Level name path
- TRIGRAM: Name&Synonym, n-Level name path, google(only wikipedia)

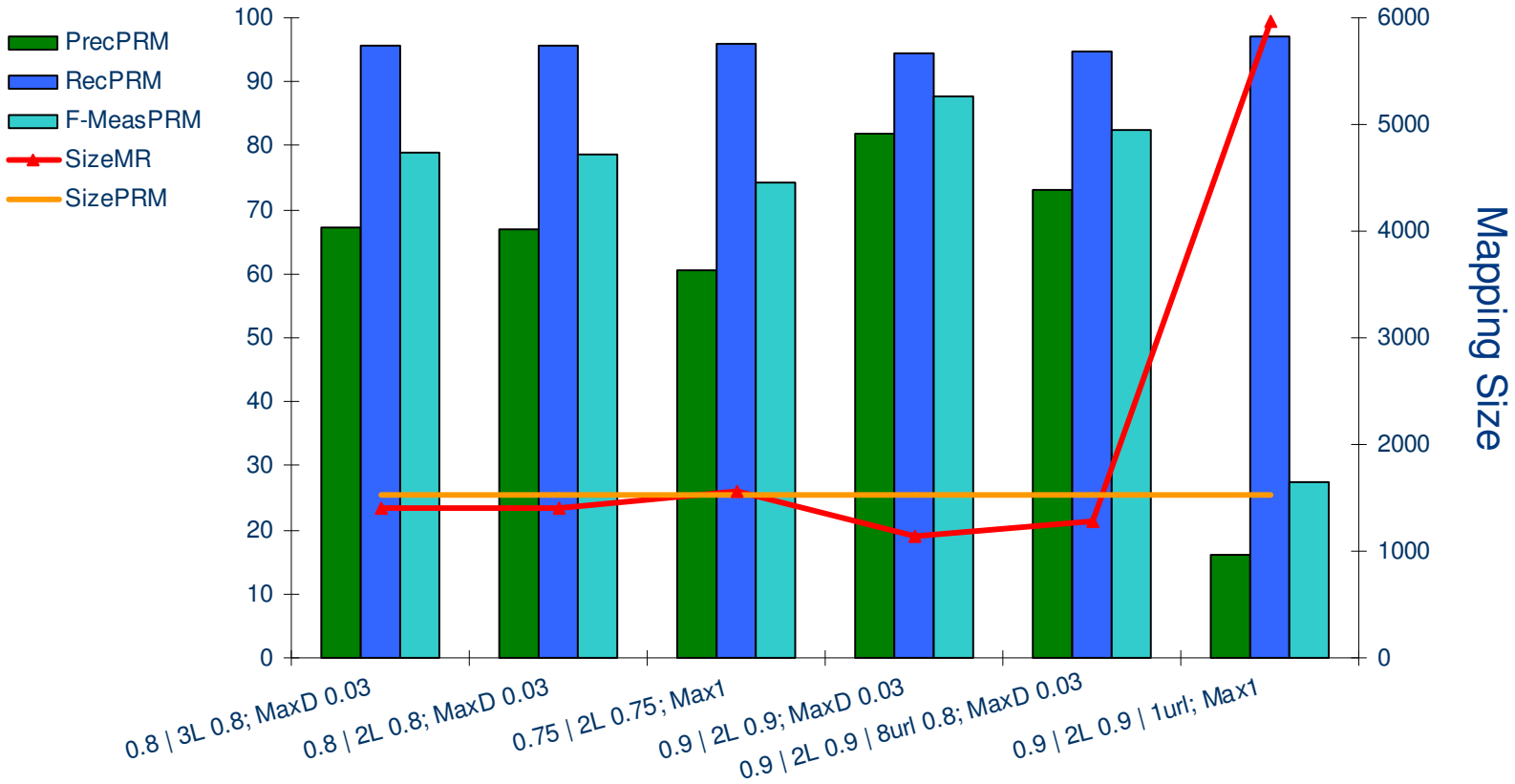
Different configurations:

- Minimal threshold
- MaxDelta parameter
- Number of levels (name path)
- #wikiURLs
- URL Overlap

Measures to evaluate quality of match results:

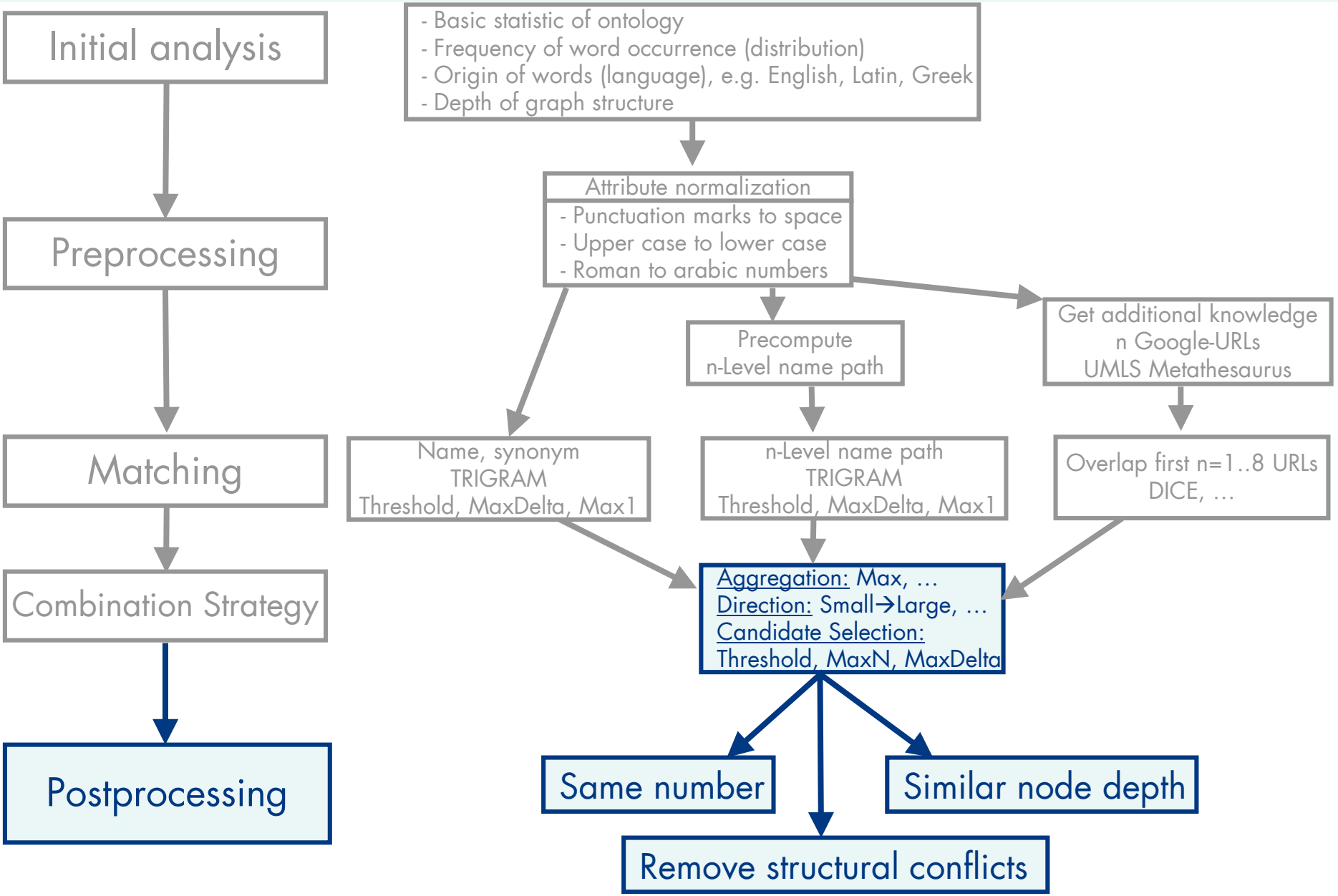
- PRM = Partial Reference Mapping
- MR = Match Result
- $Size_{PRM}$, $Size_{MR}$
- $Prec_{PRM}$, Rec_{PRM}, $F\text{-}Meas_{PRM}$ - Precision, Recall, F-Measure w.r.t. PRM

Evaluation Results



- 2- vs. 3-Level name path no big difference (use nearest semantic neighborhood)
- 0.9 (MaxDelta 0.03) to restrictive → small mapping size
  - Good if focus on precision
- wiki1 produces to much results (~6,000)
  Problem: fine and general concepts are described on the same wiki page
- wiki8 seams to be promising …

Workflow


Initial analysis
- Basic statistic of ontology
- Frequency of word occurrence (distribution)
- Origin of words (language), e.g. English, Latin, Greek
- Depth of graph structure
Preprocessing
Attribute normalization
- Punctuation marks to space
- Upper case to lower case
- Roman to arabic numbers
Precompute
n-Level name path
Get additional knowledge
n Google-URLs
UMLS Metathesaurus
Matching
Name, synonym
TRIGRAM
Threshold, MaxDelta, Max1
n-Level name path
TRIGRAM
Threshold, MaxDelta, Max1
Overlap first n=1..8 URLs
DICE, …
Combination Strategy
Aggregation: Max, …
Direction: Small→Large, …
Candidate Selection:
Threshold, MaxN, MaxDelta
Postprocessing
Same number
Similar node depth
Remove structural conflicts

Postprocessing

Todo:

- **Use different integrity constraints to filter false correspondences**
- Same numbers
  - if concepts of one correspondence contain numbers (roman or arabic) → allow only same number
- Remove structural conflicts
  - Use of semantic constraints, e.g., high-level disjointness
  - Opposite hierarchy of "as similar" identified concepts
- Require similar node depth
  - Concepts of a correspondence should have the same (a similar) distance to the root
- …

How much time took it?

Example match task:

M_1 = Trigram Name,Synonyms t=0.8 & M_2 Trigram 2-level name path 0.8

Nested loop: ~18,000,000 comparisons

- **1 CPU:**
- **11.5 min** (M_1 → 5.3 min, M_2 → 6.2 min)

Why only use one CPU out of 2/4/.. available CPUs?

- **6 CPUs:**
- **2 min (M_1 → 0.9 min, M_2 → 1.1 min)**

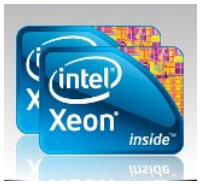
50 runs with different configurations: only ~2h instead of 9.6h

How was it realized?

Distributed Matching *

Basic idea

- Use the CPUs/main memory of available resources (e.g., in our group) to speed up ontology match task significantly
- Top database group resources

Dual-Core	Quad-Core	2x Quad-Core
		
9 Desktop clients 18 CPUs à 2.66 GHz	**2 Desktop clients 8 CPUs à 2.66 GHz**	**2 Server of db group 16 CPUs à 2.66 GHz**

Σ: 42 CPUs à 2.66 GHz

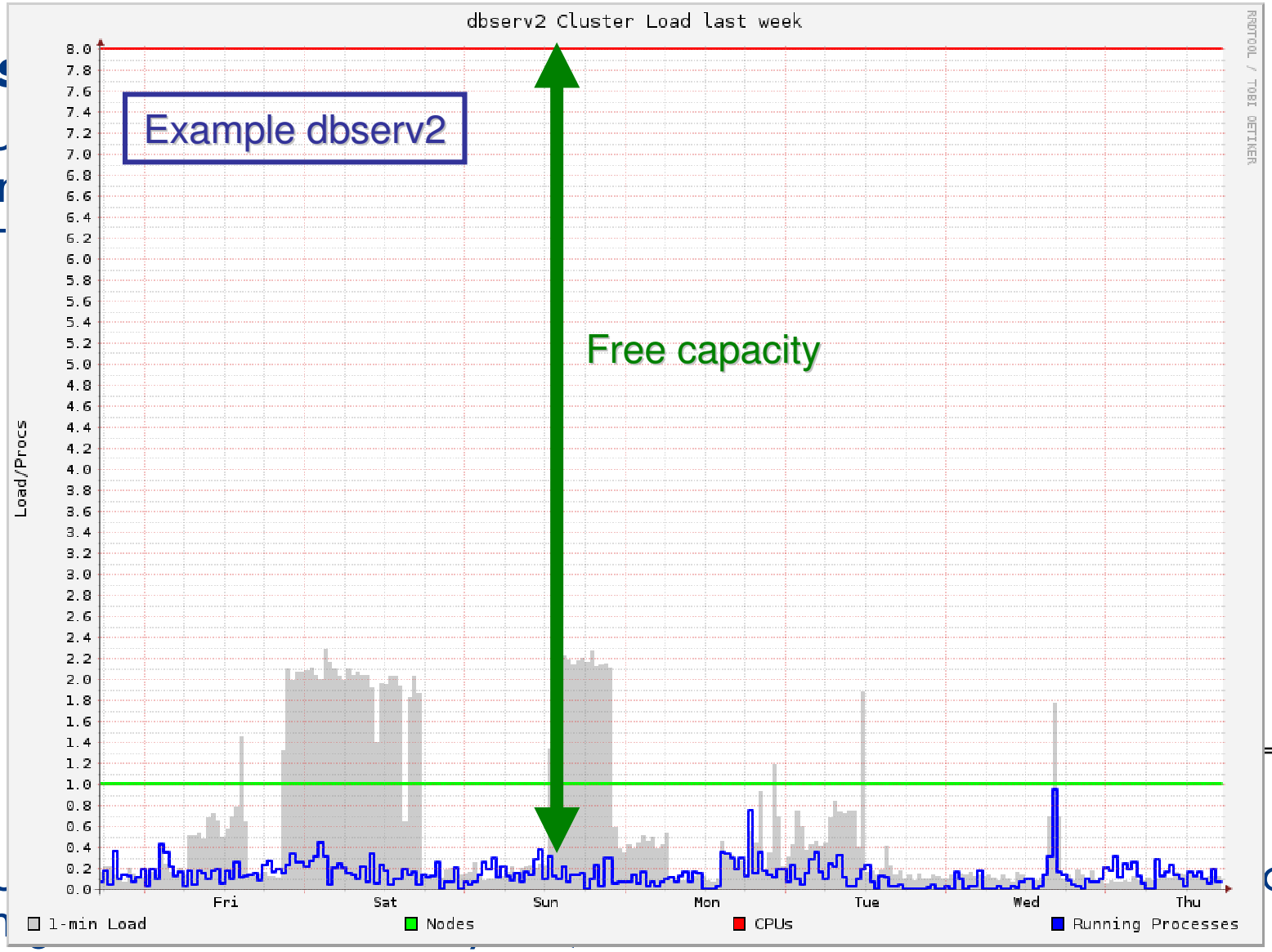
- Utilization of resources when they are not used (lunch break, over night, weekend, holiday, …)

* Joined work with Toralf and Michael

Distributed Matching *

Bas

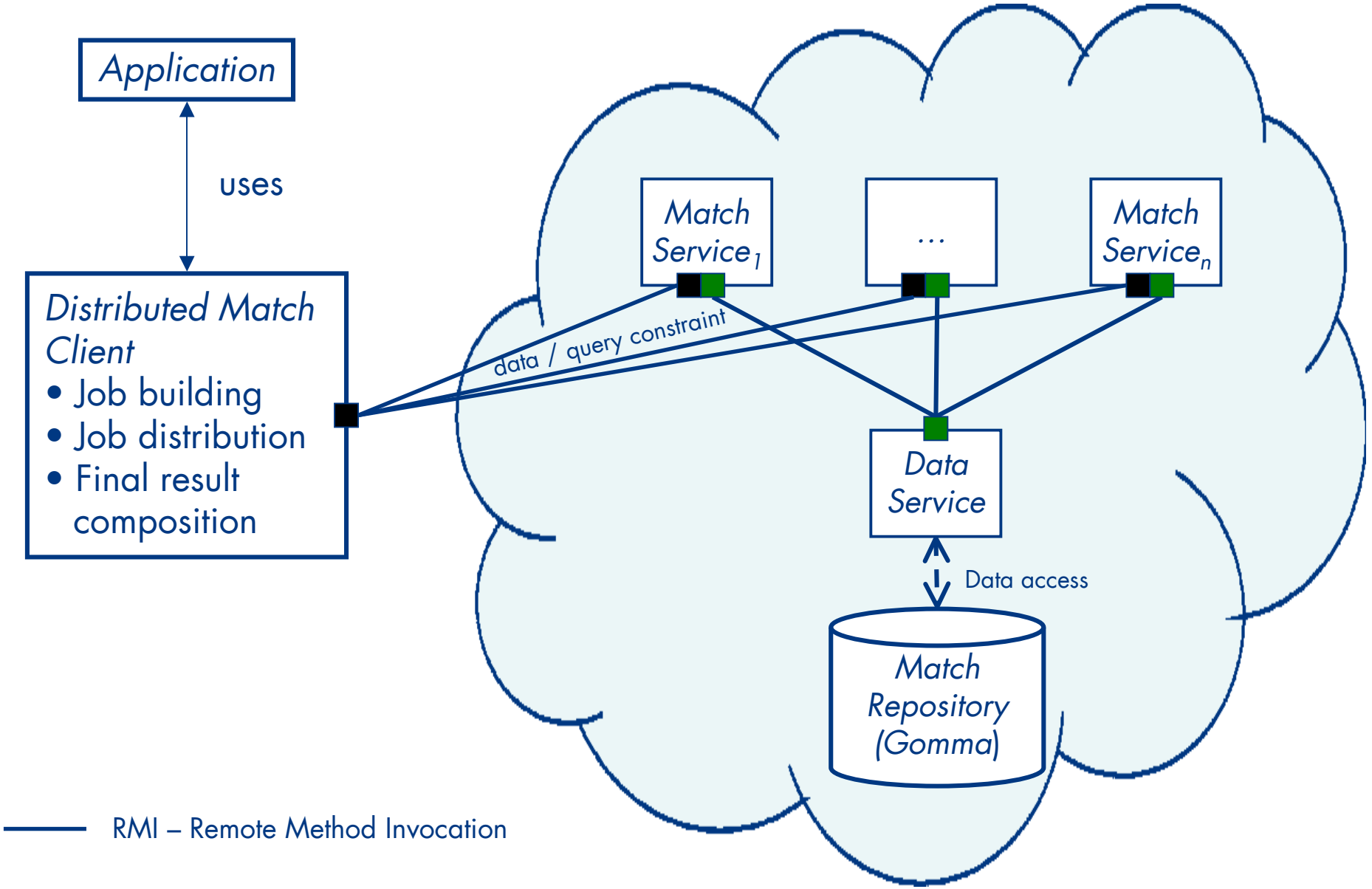
- U
  ir
- T



- U over
  n

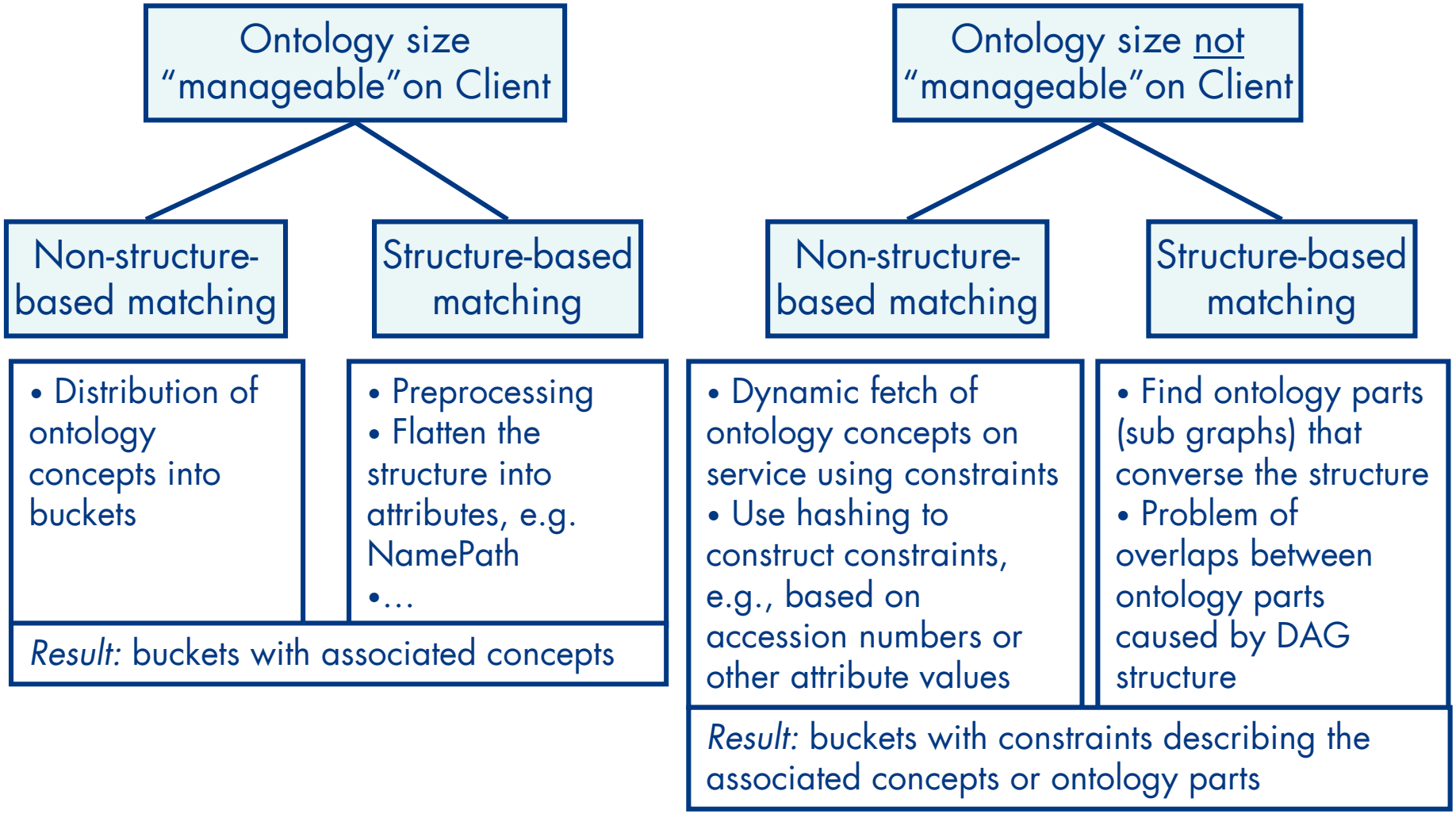
* Joined work with Toralf and Michael

Distributed Architecture



Ontology Partitioning for Distributed Matching

- Partition each ontology and match parts on services
- Merge of single results into overall result



Experiment

Experimental setup

- NCI Thesaurus self match using TRIGRAM
- 68,855*68,855 concepts = nested loop $4.8*10^9$ comparisons
- Split in 32 partitions à ~2,150 concepts on each side → 1024 jobs
- Partition: Hashing of accession number, example bucket:

  SELECT … WHERE accession LIKE „%00“
  OR accession LIKE „%01“
  OR accession LIKE „%02“

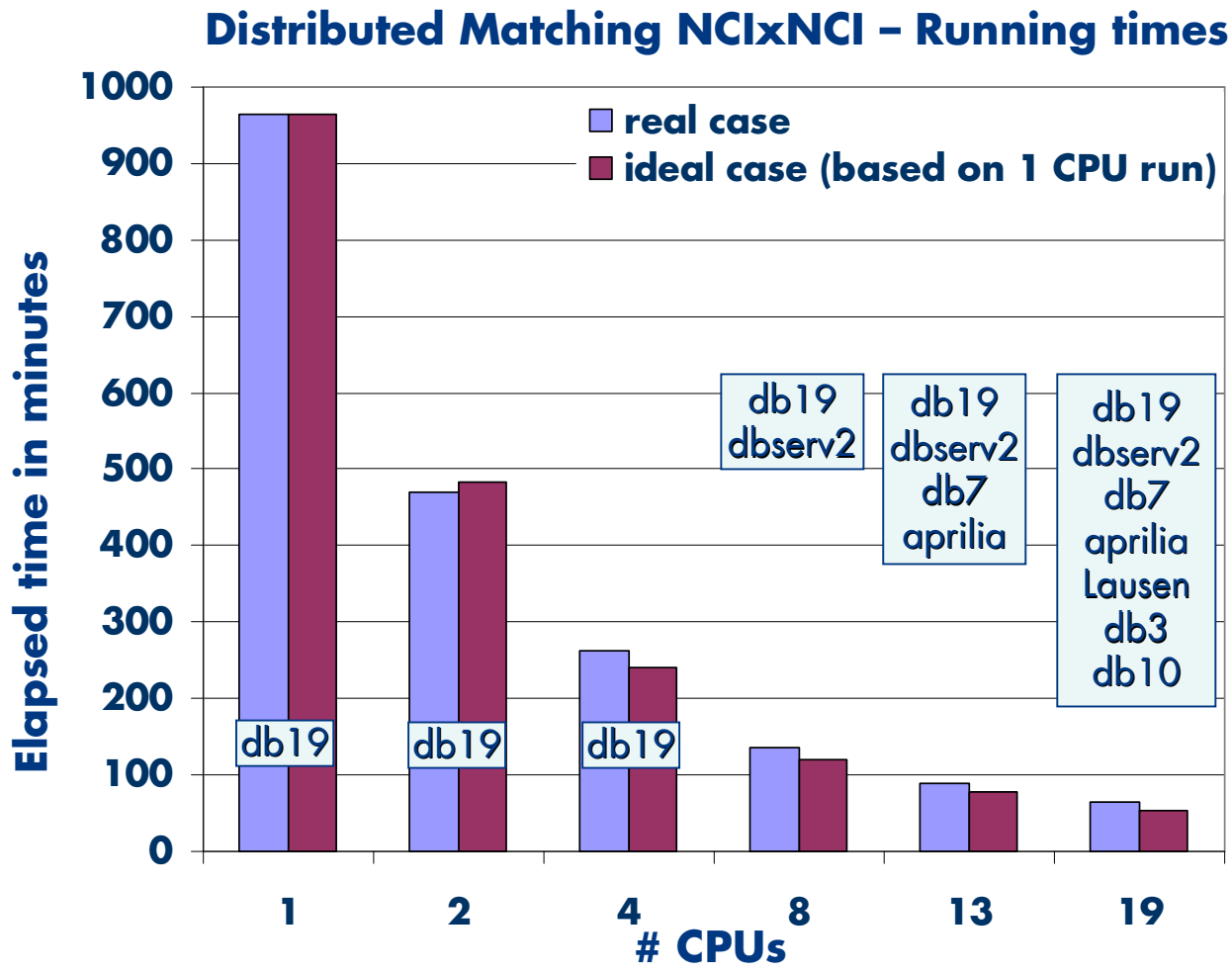
Possible to match it in the lunch break? *

- 19 CPUs from the db group* and IZBI server
- 65 minutes
- 1,223 comparisons per millisecond

How is the performance for less CPUs?

* Thanks to Stefan, Lars and Andreas for providing us their free resources ☺

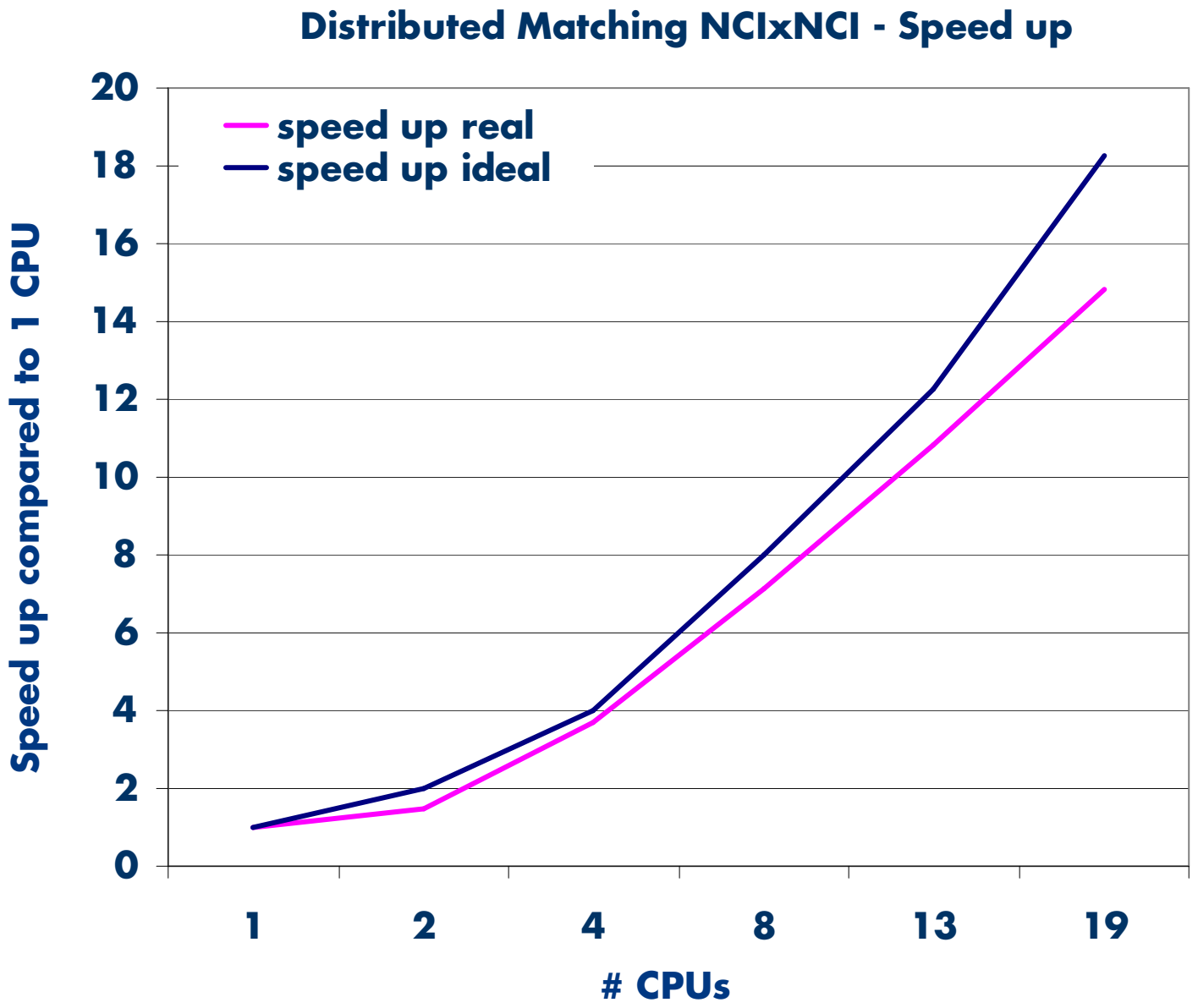
Running Times for Different #CPUs



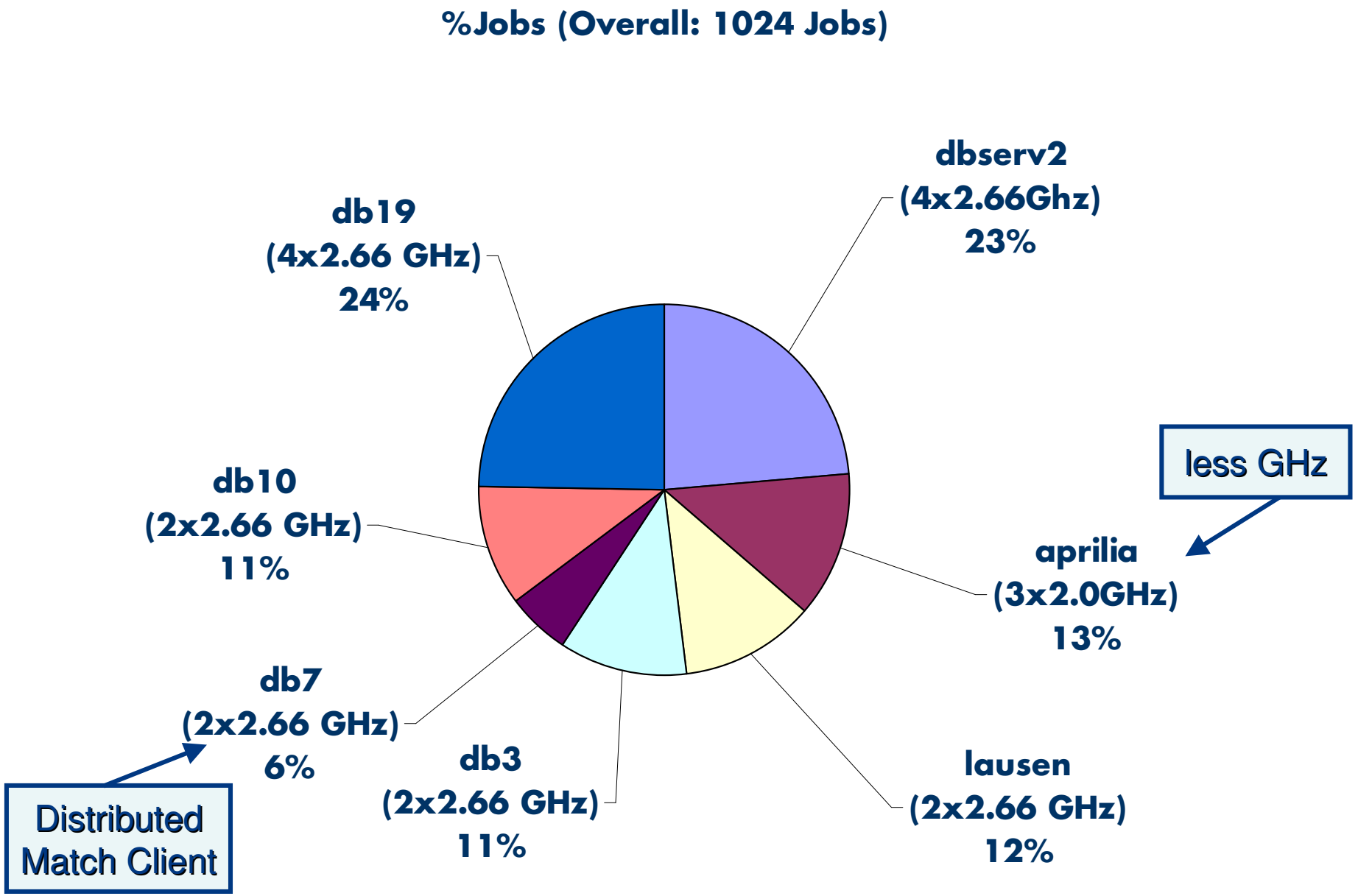
- Significant speed up even for one machine (2/4 CPUs)

→ Reduced from ~16h to ~1h

Real vs. Theoretic Speed up



Assignment of Jobs



Conclusion and Future Work

(1) Match workflow

- Experiences in aligning large life science ontologies with metadata-based strategies
- Future work
  - Improve the workflow (e.g., Google query strategies, …)
  - Evaluation of the data (send to OAEI)
  - Investigate and compare the stability/robustness of metadata-based and instance-based matching approaches

(2) Distributed matching

- Significant reduction of elapsed time for "nested-loop" matching of large ontologies/data sets
- Future work
  - Combination of distributed matching with intelligent blocking strategies
  - Interesting for object matching?

Thank you for your attention!