



Copyright (c) Google

Yevgeny Kazakov, Pavel Klinov

University of Ulm | July 22, 2013

## Experimenting with ELK Reasoner on an Android Phone

What we did and why we did it

Results

## The story

Yevgeny gets a new Google Nexus 4 for his birthday. . .

## The story

Yevgeny gets a new **Google Nexus 4** for his birthday. . .

. . . and decides to run **ELK** out of curiosity

## The story

Yevgeny gets a new **Google Nexus 4** for his birthday. . .

. . . and decides to run **ELK** out of curiosity

which was surprisingly easy since:

- ▶ it's Java, baby ©
- ▶ no 3rd party libraries (except of **log4j**)

## The story

Yevgeny gets a new **Google Nexus 4** for his birthday. . .

. . . and decides to run **ELK** out of curiosity

which was surprisingly easy since:

- ▶ it's Java, baby ©
- ▶ no 3rd party libraries (except of **log4j**)

Also, we are doing something new here:

- ▶ demonstrate what a **general** reasoner can do on a phone
- ▶ while other reasoners were developed **specifically** for phones
- ▶ or even microcontrollers!

## Why should anyone care?

Why mobile devices? There're

- ▶ desktops, laptops,...
- ▶ cloud services

## Why should anyone care?

Why mobile devices? There're

- ▶ desktops, laptops, . . .
- ▶ cloud services

However, there're still use cases for portable devices:

- ▶ **Context-aware processing:**  
phone provides sensor data (GPS, etc.),  
ontology provides background knowledge
- ▶ **Privacy:** not all personal data should be sent to the cloud



## Why should anyone care?

Why mobile devices? There're

- ▶ desktops, laptops, . . .
- ▶ cloud services

However, there're still use cases for portable devices:

- ▶ **Context-aware processing:**  
phone provides sensor data (GPS, etc.),  
ontology provides background knowledge
- ▶ **Privacy:** not all personal data should be sent to the cloud

Your phone can warm up your hands while classifying SNOMED!

## Briefly about ELK

ELK is a Java-based **concurrent** reasoner for  $\mathcal{EL}^+$

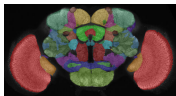
Distinctive features:

- **Concurrency**: loading/reasoning/taxonomy parallelized
- **Memory efficiency**: less GC activity on larger ontologies

Used in a number of projects:



SNOMED CT



Virtual Fly Brain



Gene Ontology

Free and open-source: [elk.semanticweb.org](http://elk.semanticweb.org)

What we did and why we did it

Results

## Experimental Setup

Google Nexus 4



Qualcomm Snapdragon S4 Pro  
4 cores, 1.7 GHz, 2 GB RAM

Personal laptop



Intel Core i5-2520M  
4 cores, 2.5Ghz, 8 GB RAM

## Experimental Setup

Google Nexus 4



Qualcomm Snapdragon S4 Pro  
4 cores, 1.7 GHz, 2 GB RAM

Personal laptop



Intel Core i5-2520M  
4 cores, 2.5Ghz, 8 GB RAM

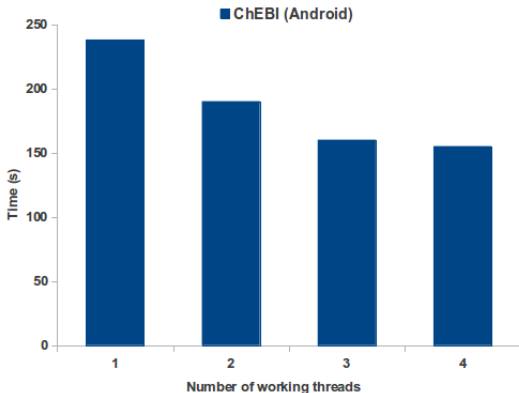
	Ontology	Size
ChEBI	Chemical Entities of Biological Interest	67,182
EMAP	e-Mouse Atlas Project Ontology	13,730
Anatomy	the Fly Anatomy Ontology	19,137
GO	Gene Ontology (old)	28,896
EL-GALEN	$\mathcal{EL}^+$ version of GALEN	36,547

## Results

For ChEBI (67,182 axioms):

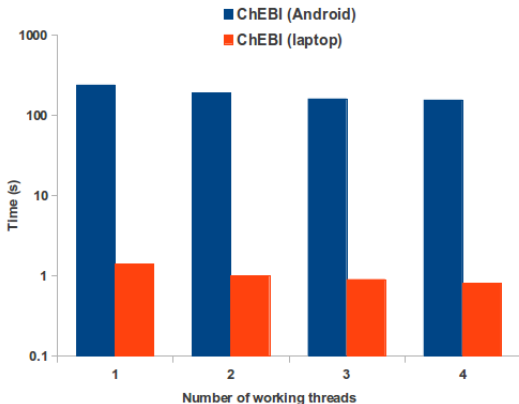
## Results

For ChEBI (67,182 axioms):



## Results

For ChEBI (67,182 axioms):





## Results

SNOMED CT completed in **<1 hour** (consuming all 500GB)

## Summary

Well-engineered reasoners can work on mobile devices

The same optimizations work:

- ▶ concurrency
- ▶ reducing memory footprint
- ▶ efficient representation of  $\mathcal{EL}$  rules

## Summary

Well-engineered reasoners can work on mobile devices

The same optimizations work:

- ▶ concurrency
- ▶ reducing memory footprint
- ▶ efficient representation of  $\mathcal{EL}$  rules

Performance difference **not** proportional to difference in computational power

- ▶ different GC, JIT, ...
- ▶ slower IO
- ▶ slower/smaller CPU caches

Questions?