Scalable, Flexible and Generic Instant Overview Search

Pavlos Fafalios, Ioannis Kitsos and Yannis Tzitzikas

Presentation by: Pavlos Fafalios

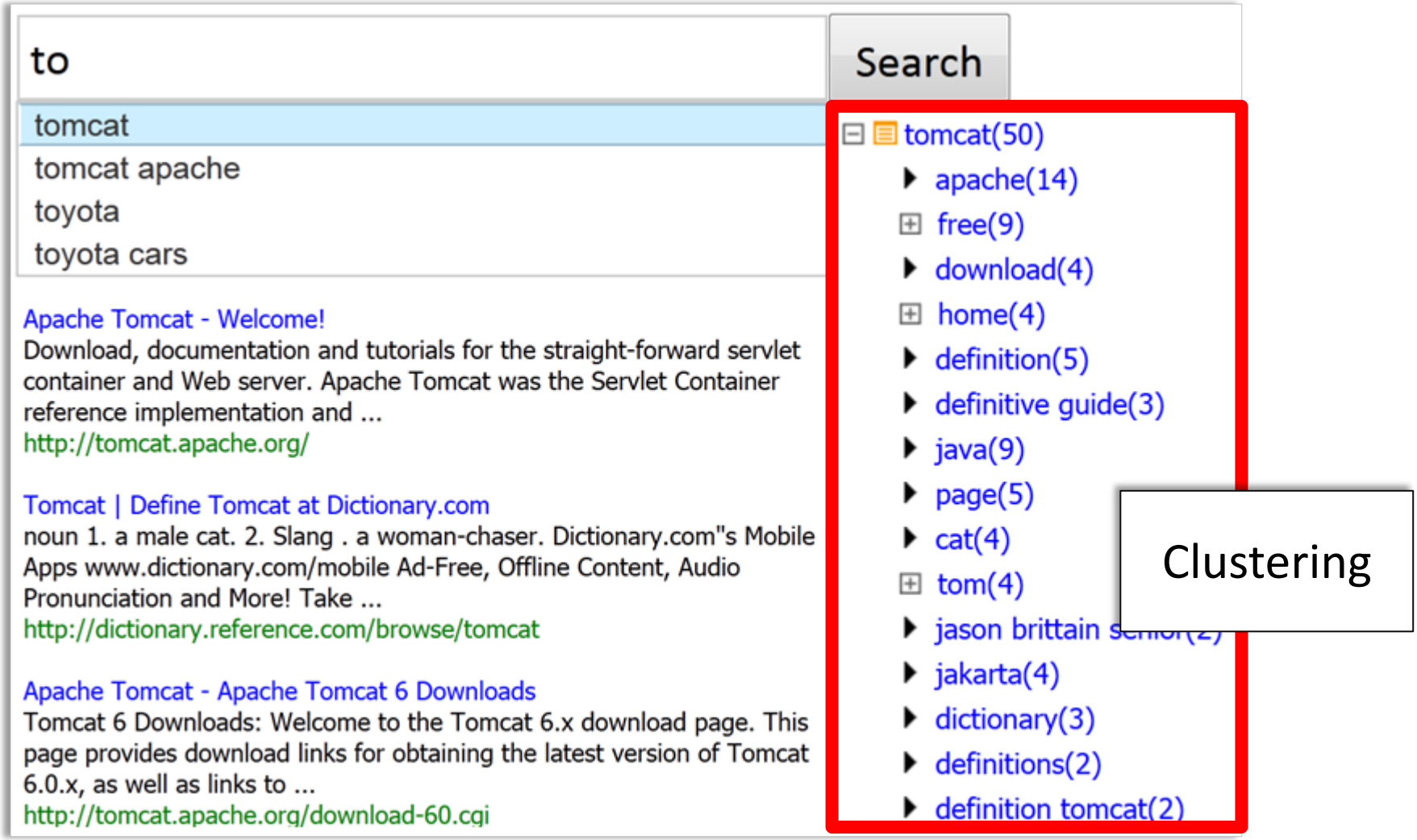
(fafalios@csd.uoc.gr)

FORTH-ICS
Information Systems Laboratory

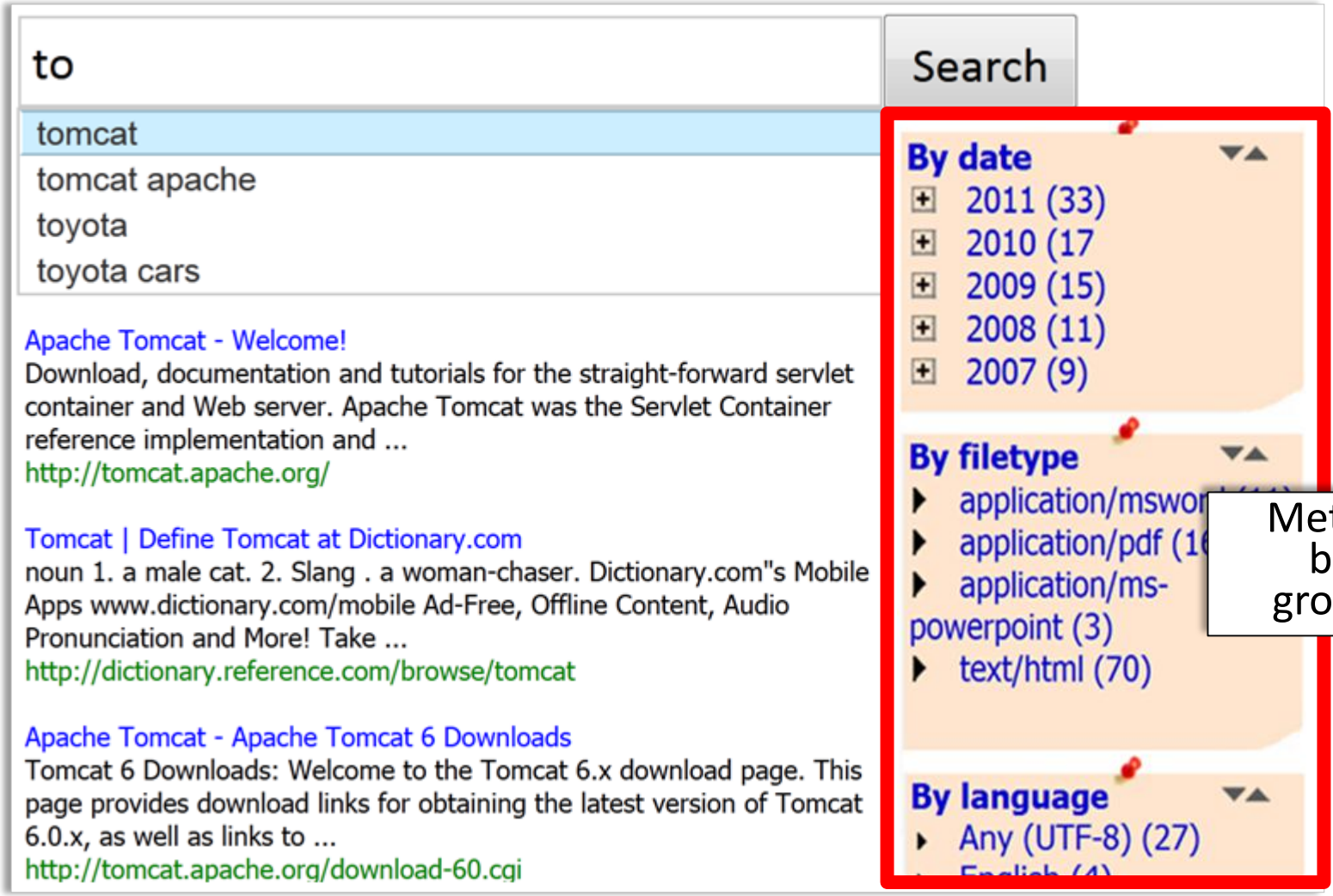
University of Crete
Computer Science Department

What is Instant Search?

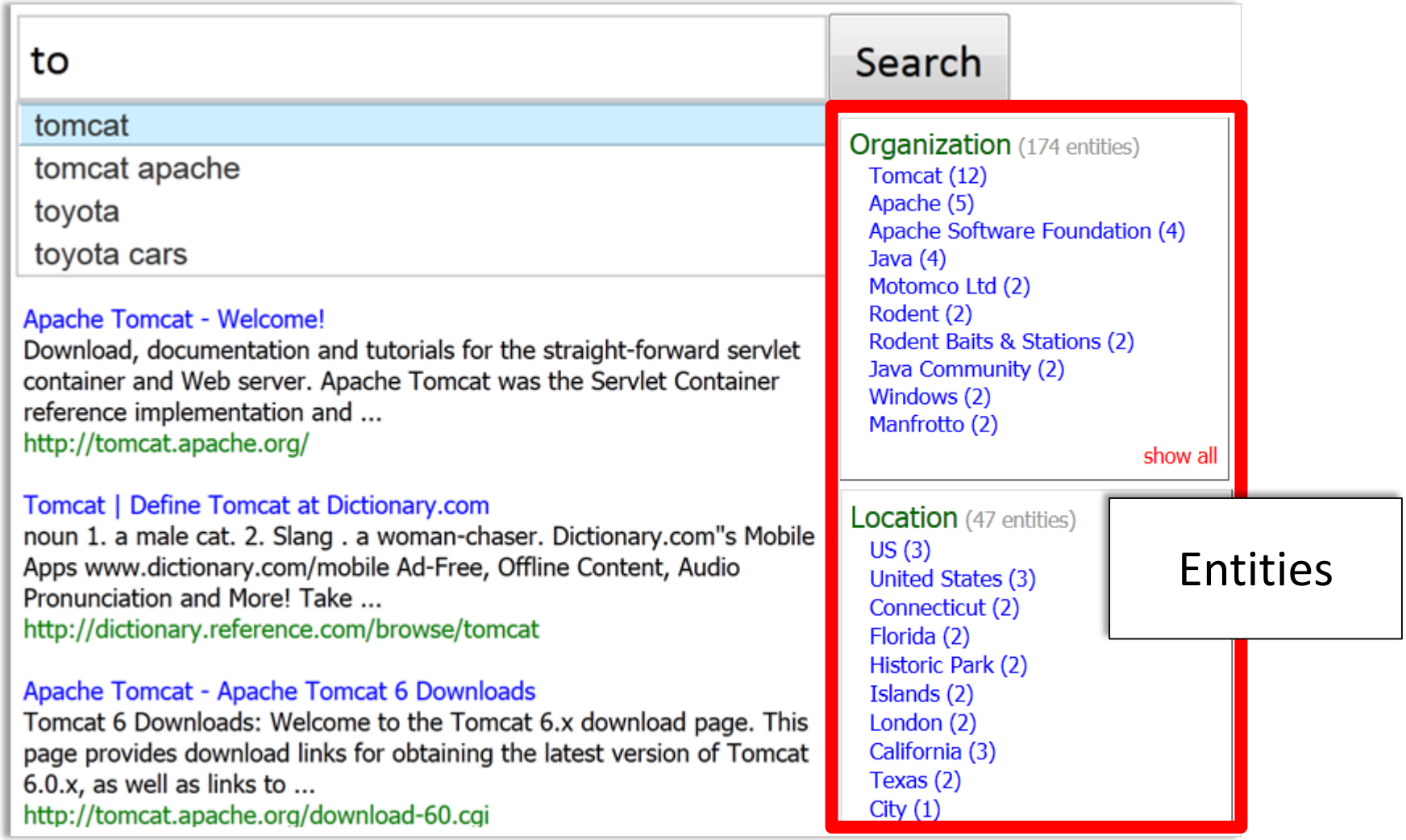
What is **I**nstant **O**verview **S**earch (**IOS**)?



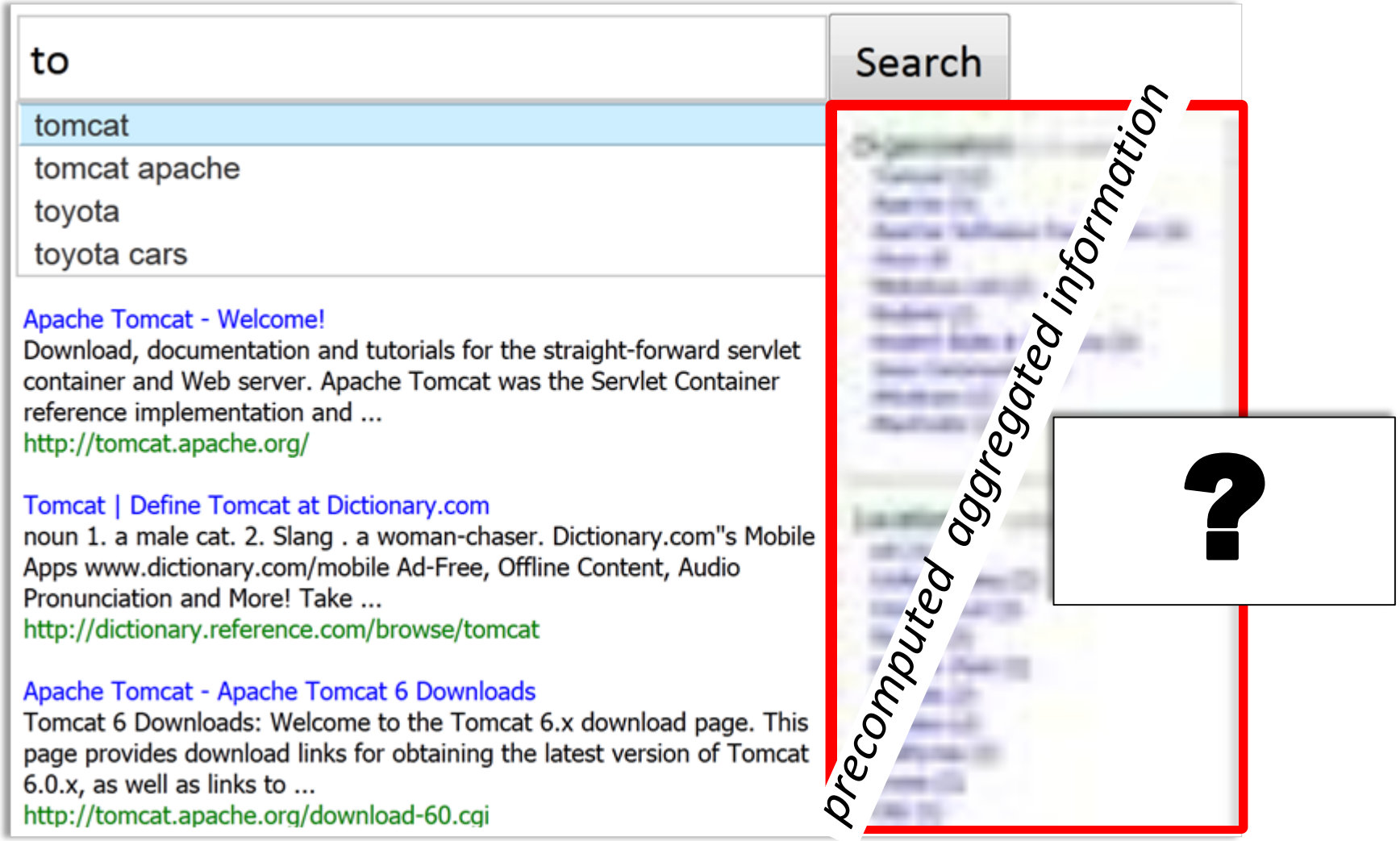
What is **I**nstant **O**verview **S**earch (**IOS**)?



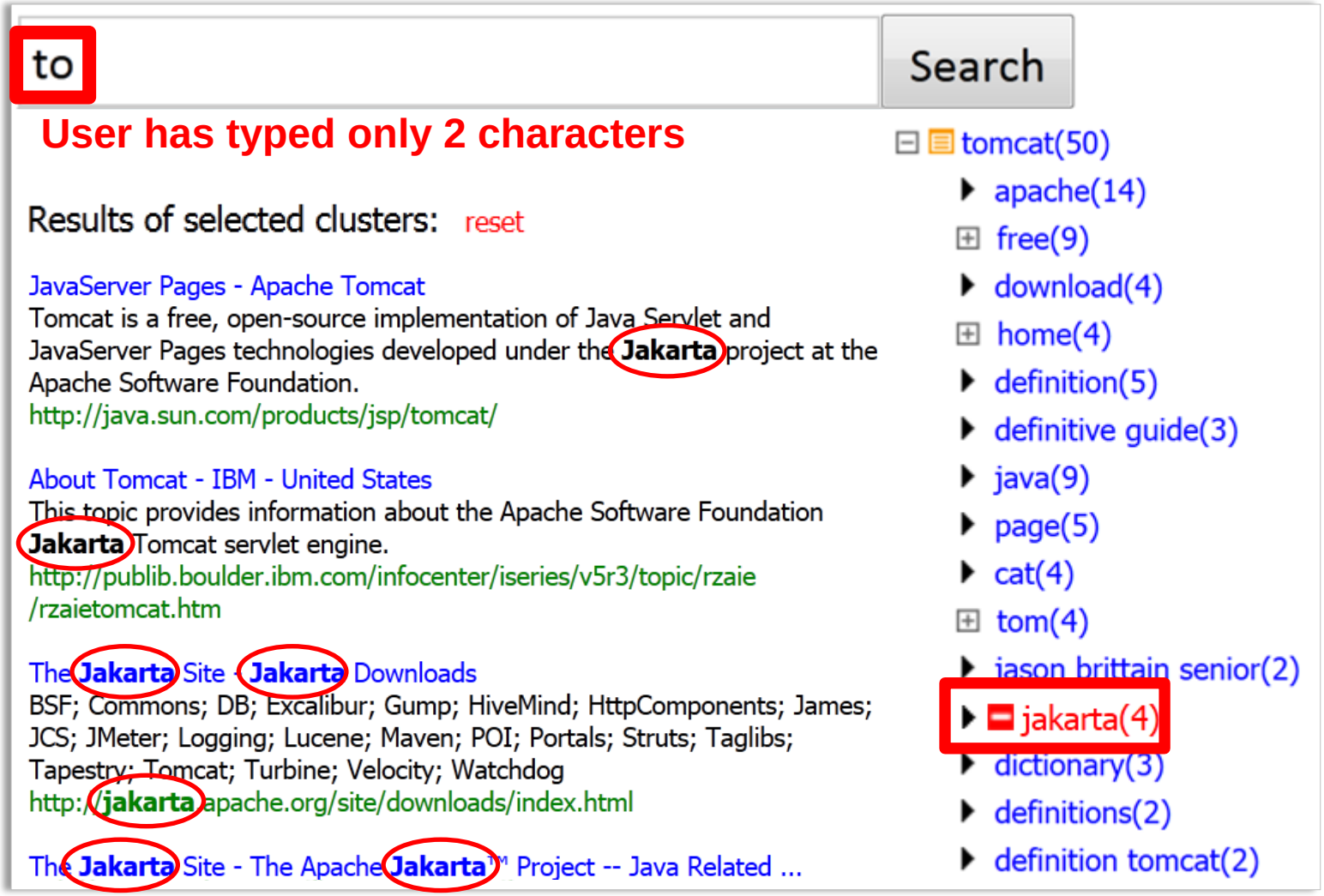
What is **I**nstant **O**verview **S**earch (**IOS**)?



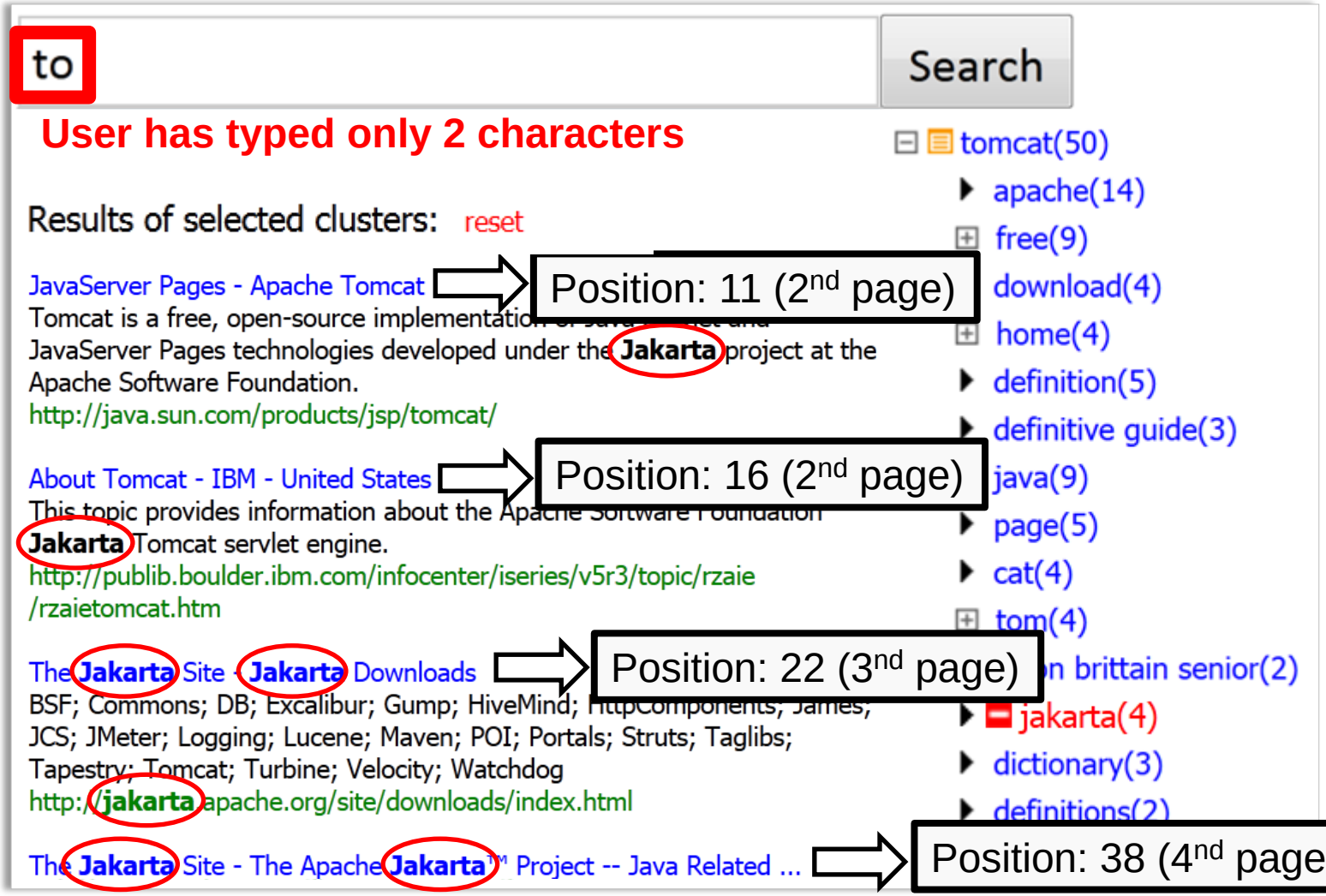
What is **I**nstant **O**verview **S**earch (**IOS**)?



What is **I**nstant **O**verview **S**earch (**IOS**)?



What is **I**nstant **O**verview **S**earch (**IOS**)?



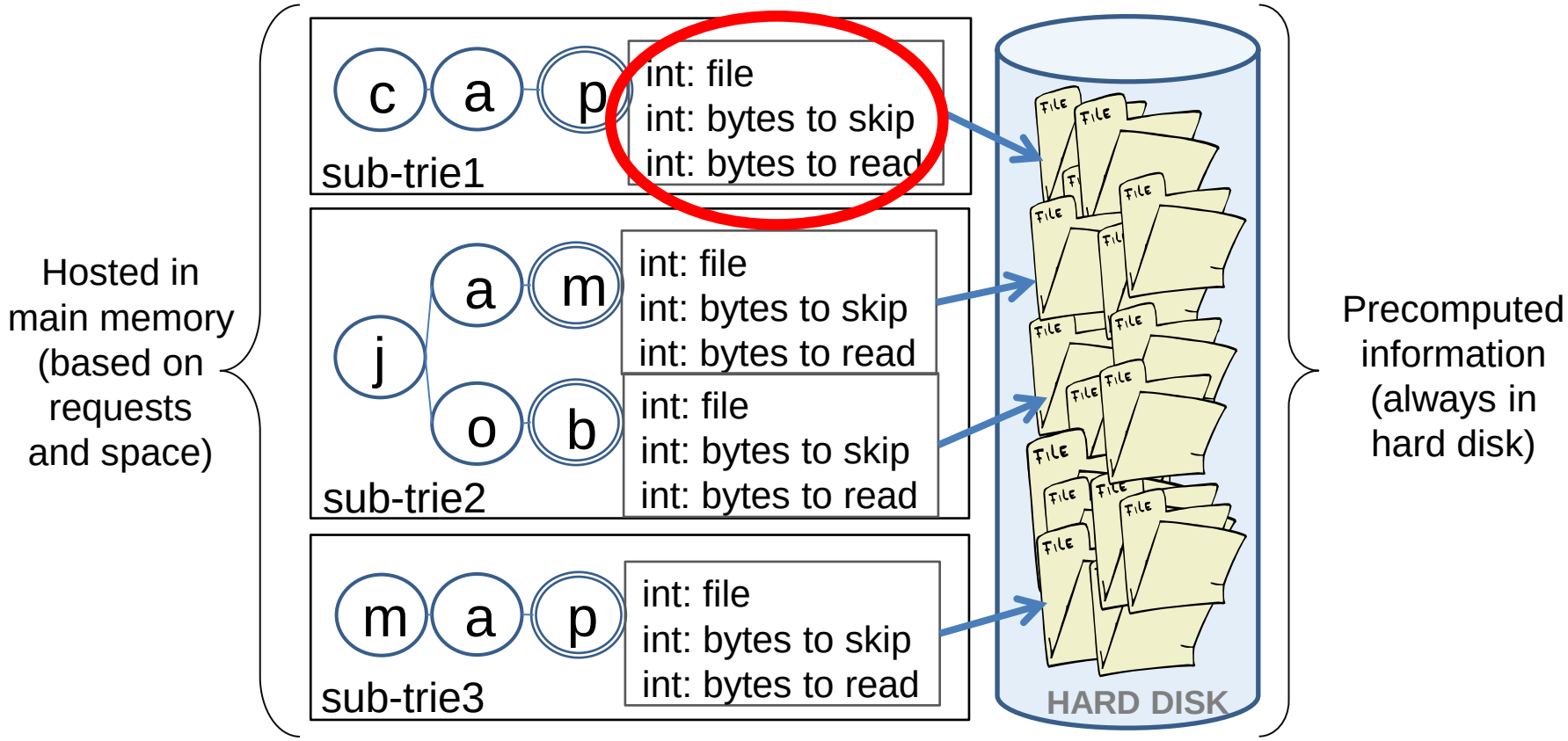
Key Challenges

- Efficiency and Scalability
  - *Offer real-time interaction*
  - *Exploit available main memory and disk*
- Flexibility
  - *Reduce user's effort*
- Generality
  - *Ability to accommodate various kinds of supplementary information*

Using very modest hardware!

(server running on an ordinary personal computer)

Our approach: *Partitioned Trie-based Index*



Average Retrieval Time ≈ **135ms**

Experiments** over a server running on a **modest personal computer**, with a synthetic query log of **1 million** distinct queries and synthetic precomputed information of **1 Terabyte

Our approach: *Hybrid Caching Scheme*

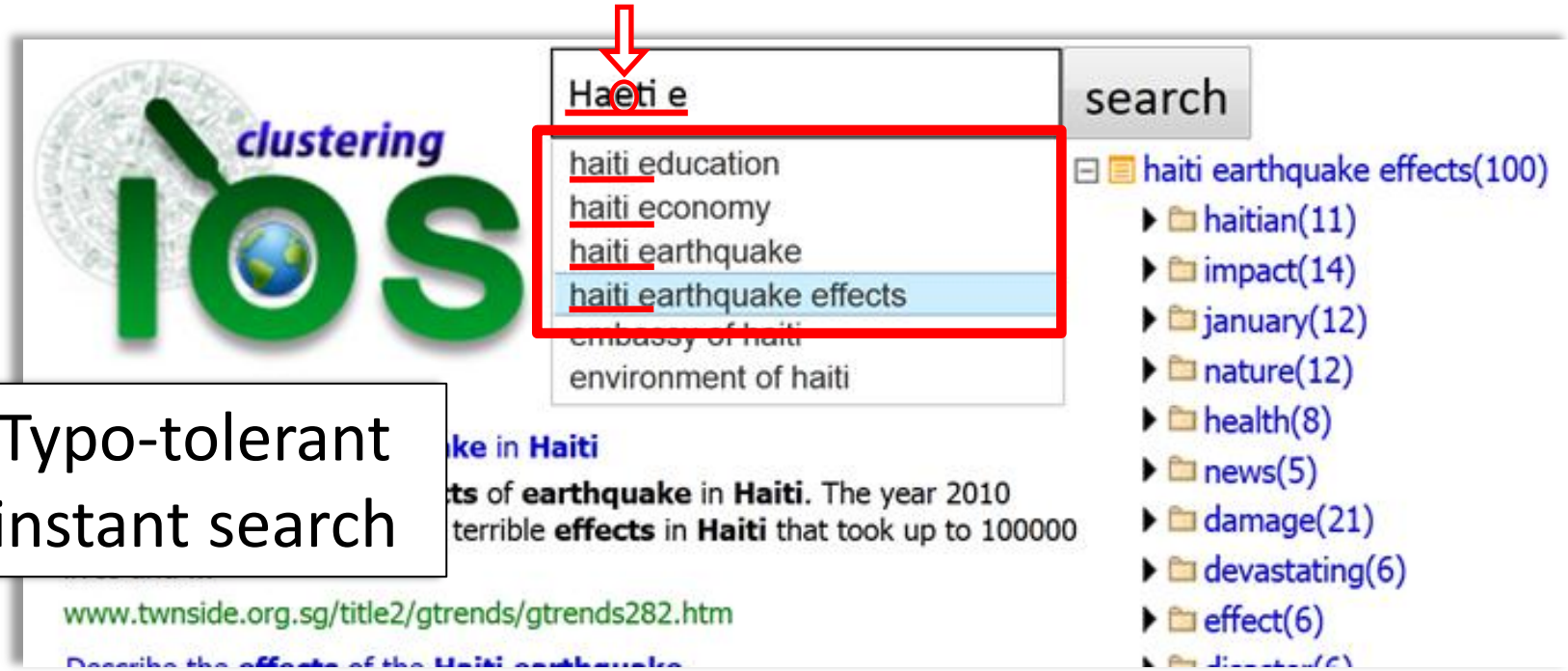
- The problem:
  - A large number of users start typing queries at the same time.
  - The system requires loading multiple subtries, i.e. the appropriate subtrie for each user's keystroke
  - **The system may get overloaded!**
- Solution: **Adopt a caching scheme**
  - Keep in memory a number of subtries
  - Options: **full static** cache, **full dynamic** cache, **hybrid** cache
- A hybrid caching scheme (70% static and 30% dynamic) offers*:

80%
better throughput

25% speedup
for queries that lie in the index

Experiment:** Synthetic query log of **1 million** distinct queries and synthetic precomputed information of **1 Terabyte

Our approach: *Tolerate Typos and Different Word Order*



Typo-tolerant instant search

Our approach: *Tolerate Typos and Different Word Order*

Word order independent instant search

Current IOS Prototypes

1. Meta-search engine (over Google) offering instant **clustering** of the snippets of the top-100 results.

2. Standalone web search engine offering instant **meta-data based groupings** of the search space.

3. Meta-search engine (over Google) offering instant **entity mining** of the contents of the top-50 results.

4. Meta-search engine (over Google) offering both instant **entity mining** of the contents and instant **clustering** of the snippets of the top results.

Conclusion

- **IOS:** A more powerful *search-as-you* type functionality.
  - Predicts our search and shows results and **supplementary information** before finish typing

- The overall **effectiveness** of IOS, apart from its efficiency, depends on the **quality** of the precomputed information.
  - However, IOS can be applied over **ANY** *autocompletion*, *ranking*, *clustering*, *entity mining*, *etc*., algorithm

Further Research

- Measure the effect of IOS to user satisfaction
  - We already gather user feedback by keeping a detailed log
- How exactly users exploit the precomputed information that appears instantly
  - This requires very fast eye-tracking equipment

Running prototypes:

http://www.ics.forth.gr/isl/ios

Thank you!