Huazhong University of
Science and Technology

華中科技大學

Fast Snippet Generation Approach Based On CPU-GPU Hybrid System

Ding Liu, Ruixuan Li, Xiwu Gu,
Kunmei Wen, Heng He, Guoqiang Gao

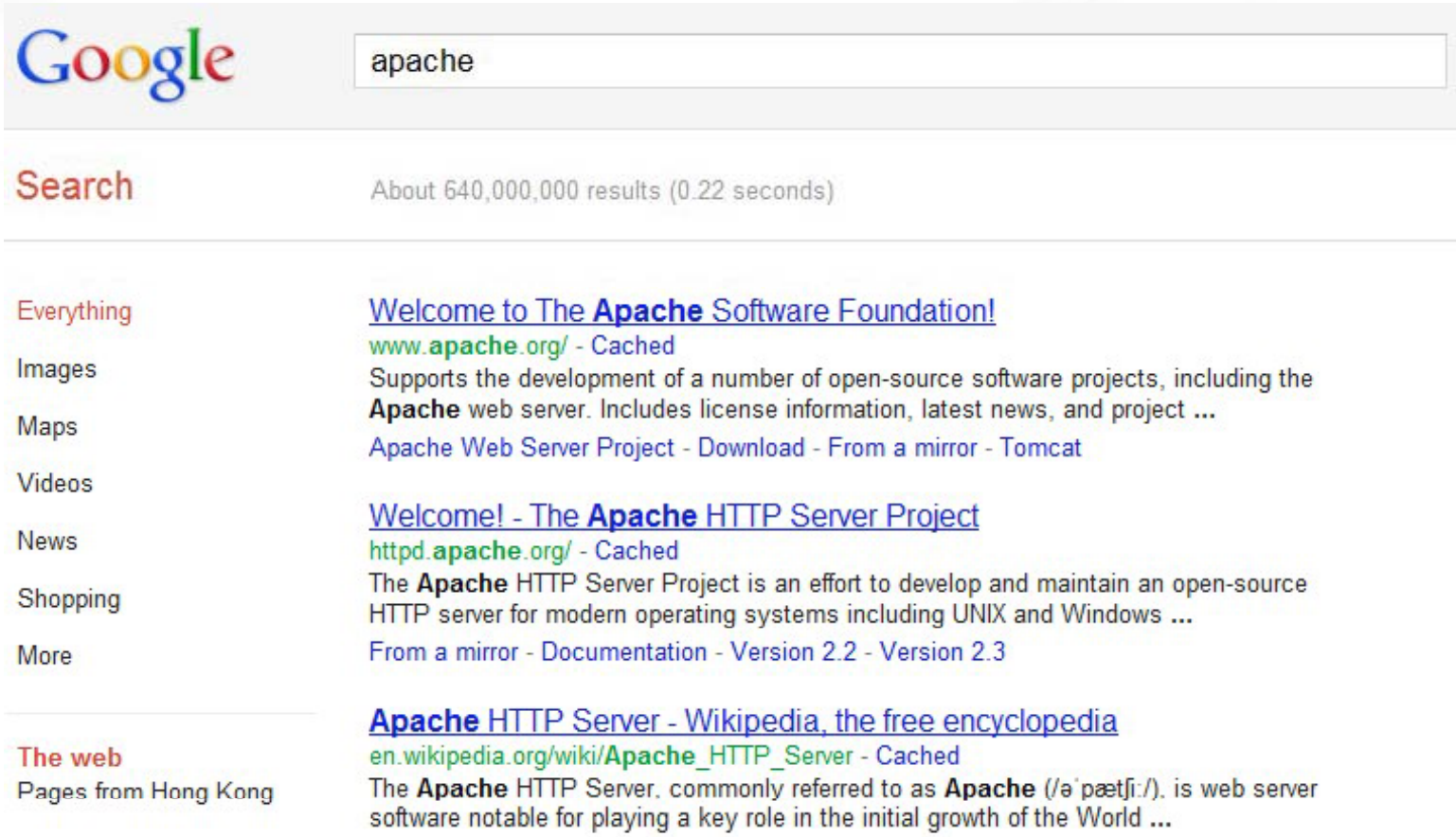
Huazhong University of Science and Technology,
Wuhan, China

Outline

- Background
- Motivation
- Approach
- Experiments
- Conclusion

What is Snippet?

Snippet is a set of fragments which are extracted from the source document.



Characteristics of Snippet

- Snippets help users understand the relevance of the documents and the query.
- The workload of snippet generation is very huge.
- As different queries on a document will generate different snippets, we cannot take the document as the key to cache the snippets.

Snippet Generation Approach

- Query-independent approach (static)
  - Generate snippet without query, and the snippet is always the same for a certain document.
- Query-biased approach (dynamic)
  - Generate snippet with query, and the snippet of a certain document is different when using different query.

Query-biased Snippet Generation

- Document-based approach
  - Each < document, query > pair as an isolate process unit.
  - Easy to parallelize
- Index-based approach (recursive)
  - Holger Bast , et al. WWW 2009
  - Using inverted index lists to intersect recursively.

Challenges

- Highlighter is an implementation of document-based approach that is widely used for snippet generation in enterprise search engine applications.
- There are two defects in this approach.
  - It adopts a serial computational model and this may lead to low efficiency.
  - The high relevant fragment may be cut off and will cause lower snippet precision.

Consideration of Snippet Generation

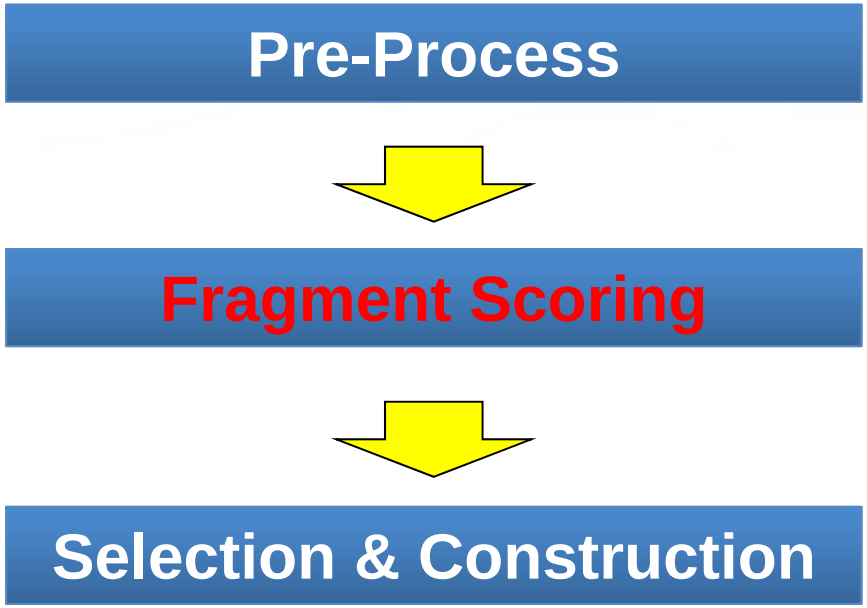
- Lightweight task
- Computation intensive
- Hard to cache

Snippet Generation: Massive small tasks, but the processes are the same.

Contributions of this Paper

- Design a process stream for snippet generation application of search engines using CPU-GPU hybrid architecture.
- Use sliding window for document segmentation.
- Parallelize the process of snippet generation for each segmentation and scoring.

Snippet Generation Process



Document segmentation and fragment filtering.

Calculate the relevance of each fragment against the query

Select the top relevant fragments as output snippet

Document Segmentation

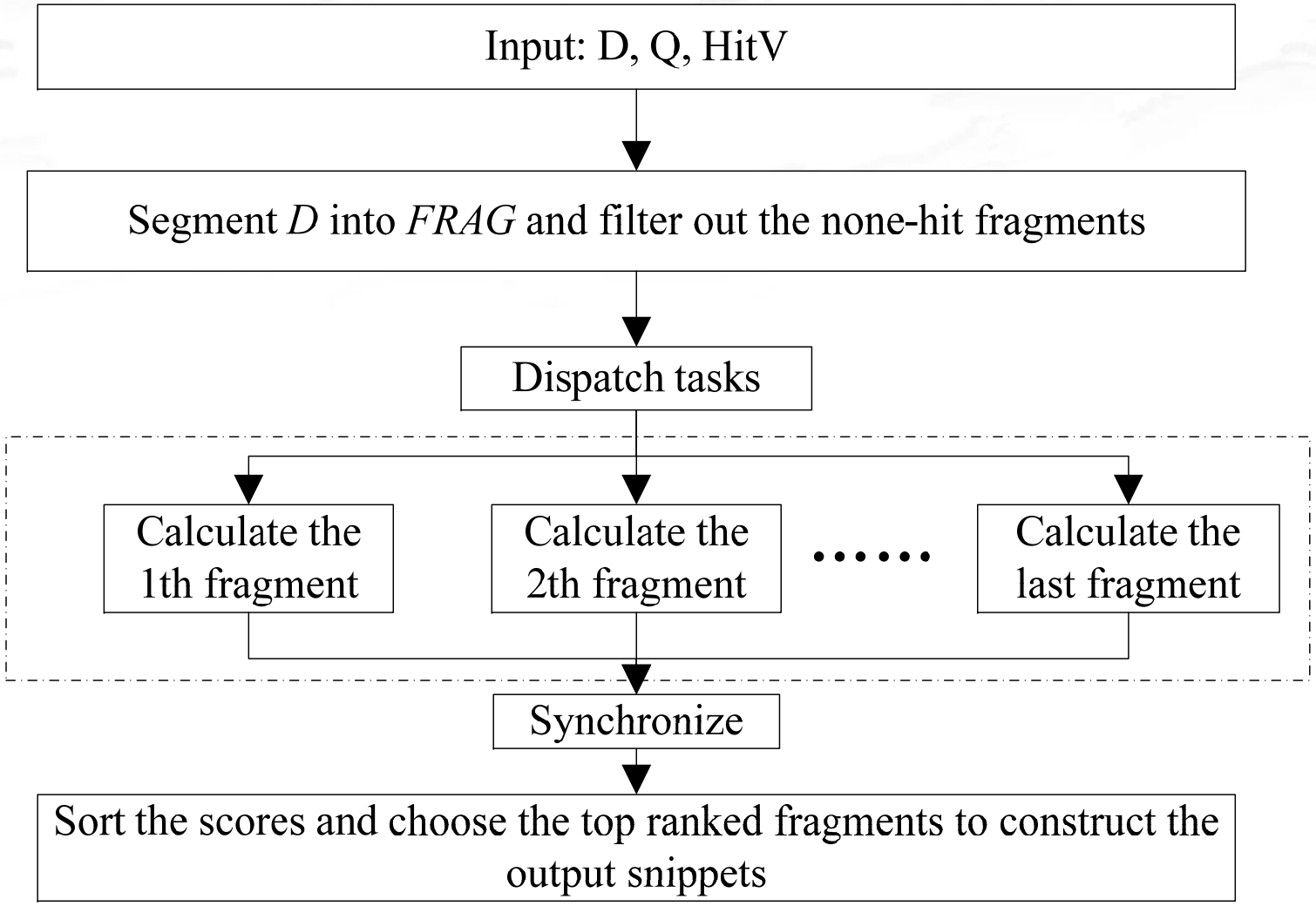
- **Two types of document segmentation**
  - Truncating segmentation: cut the document to fragments with the same length.
    - The drawback of this method is that high relevant fragment may be cut off in the process.
    - Compared to sliding method, the amount of fragments is smaller, and it does good to performance.
  - Sliding segmentation: segment a document as a sliding window, and all the fragments have the same length.
    - This method will not cut off the high relevant fragment, but it will generate much more fragments than the truncating method.
    - Besides, we should filter out the useless and duplicate part.
    - We propose using GPU to deal with more fragments.

Fragment Scoring

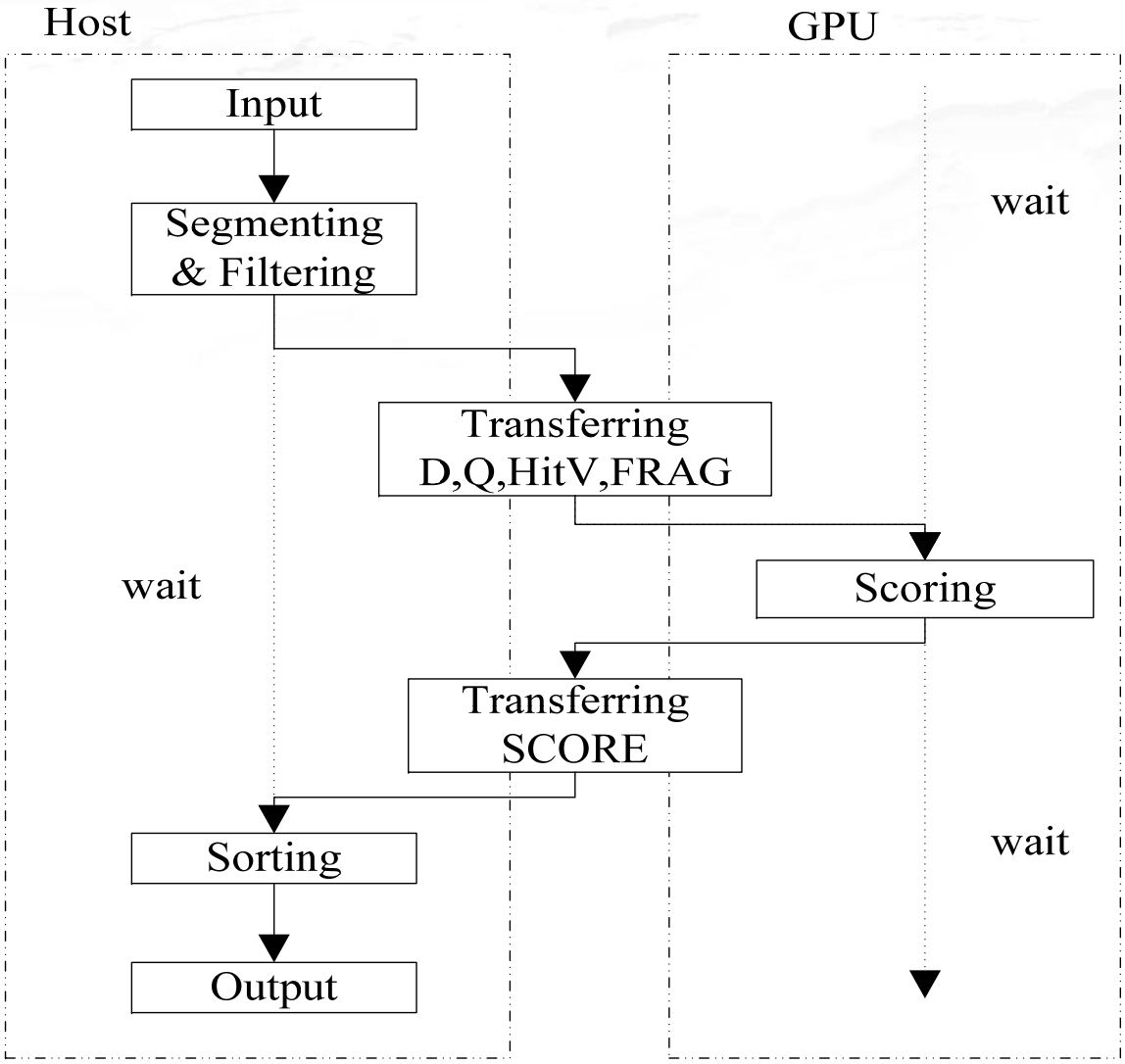
- The common method to evaluate the relevance of a document and a query is based on support vector machine (SVM).
- We consider a fragment as a small document. As the fragment is much more shorter than a document, we simplify the calculation as follows.

$$\mathbf{ScoreFun}(\mathbf{Q},\mathbf{F}) = \mathbf{coord}(\mathbf{Q},\mathbf{F}) \times \mathbf{boost}(\mathbf{F}) \times \sum_{\mathbf{term\ t\ in\ Q}} \mathbf{tf}(\mathbf{t\ in\ F}) \times \mathbf{idf}(\mathbf{t})^2 \times \mathbf{boost}(\mathbf{t})$$

Architecture



Interaction between CPU and GPU



Pipeline

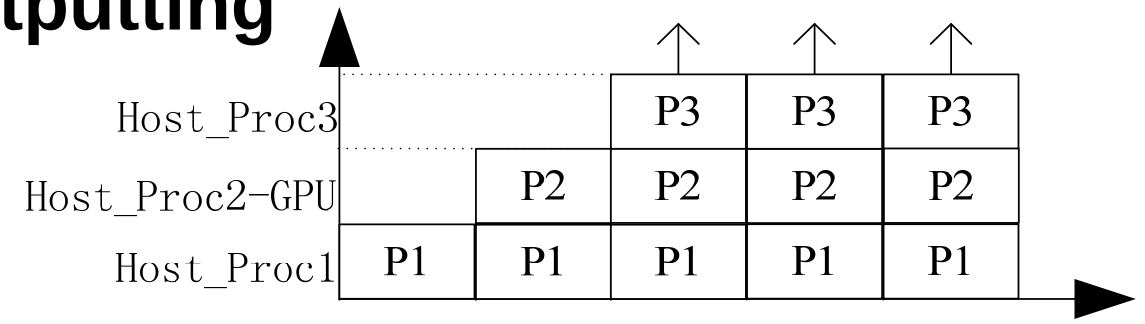
P1: Preprocessing

- Segment the documents into fragment sets, and filter out the irrelevance fragments

P2: Transferring & scoring

- Transfer D, Q, HitV to GPU
- Score in GPU
- Transfer back to host memory

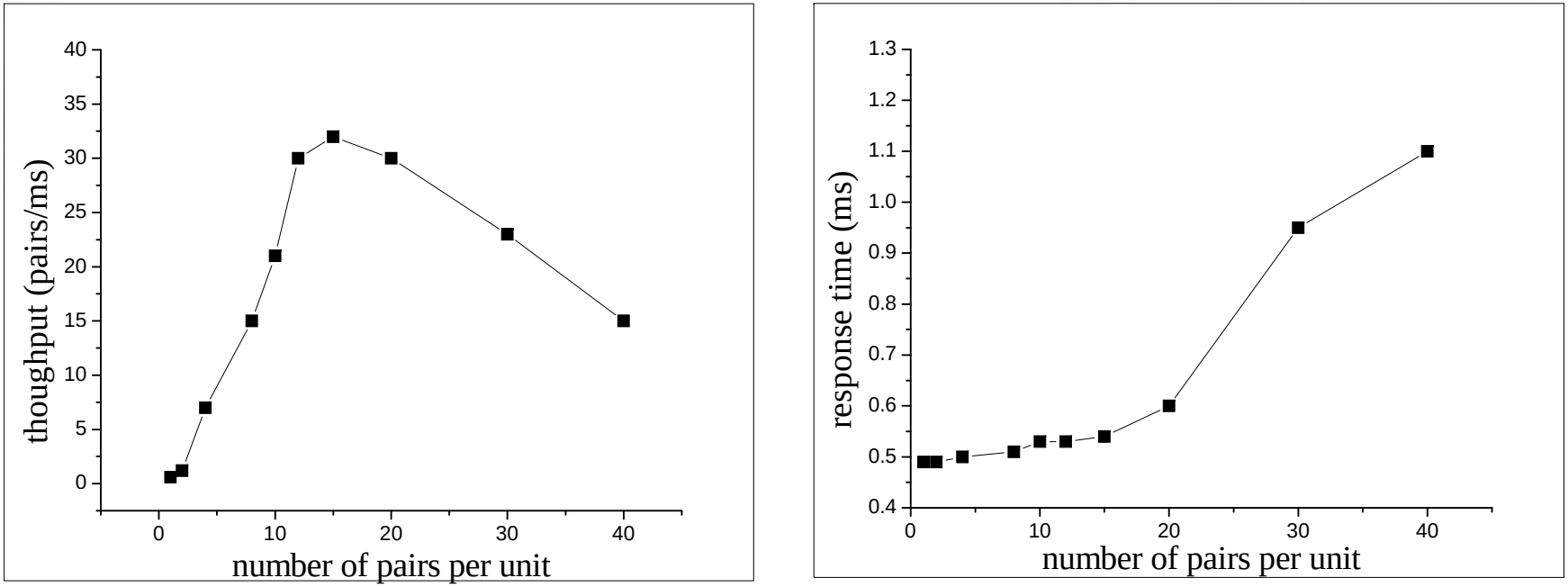
P3: Sorting & outputting



Experiment settings

- Intel Core2-Duo processors (2.2 GHz and 1 MB cache each) and 4 GB of main memory.
- GPU device is NVIDIA GTX200 / GTX250.
- All the codes are implemented in C++ and CUDA.
- The corpus was selected from Reuters Corpus English Language 2006, 2007 and 2008.
- The queries were selected from 2006 TREC efficiency topics.
- Data set containing 3,000,000 document-query pairs.

Performance with different process unit size

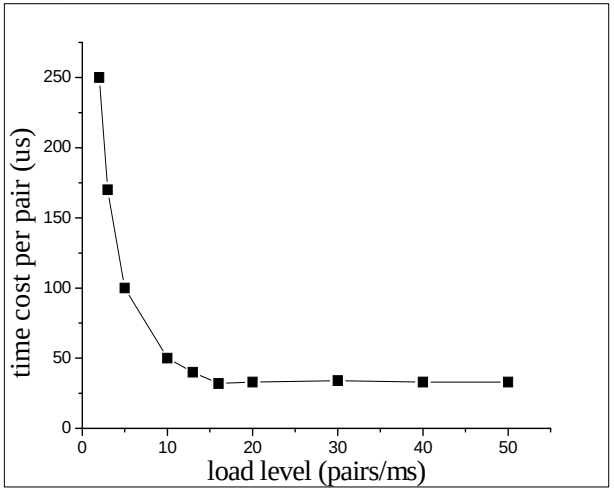


Throughput and response time with different process unit size

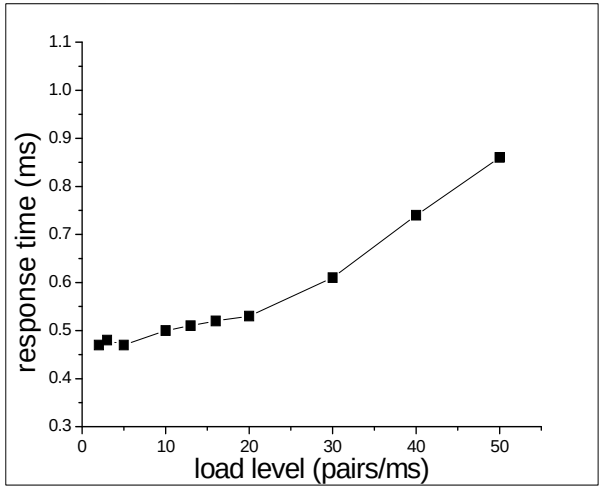
- We adopt 15 as the process unit size in the following experiments.

Efficiency under different load levels

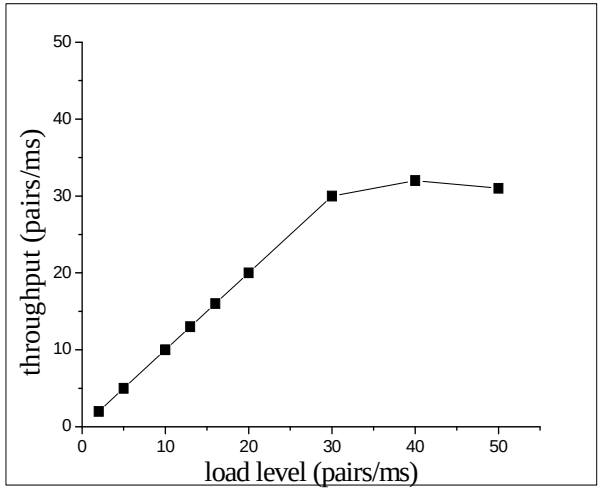
- We tested the pipeline under different pressure to find out the performance under different work loads.



(a) Time consuming of a pair



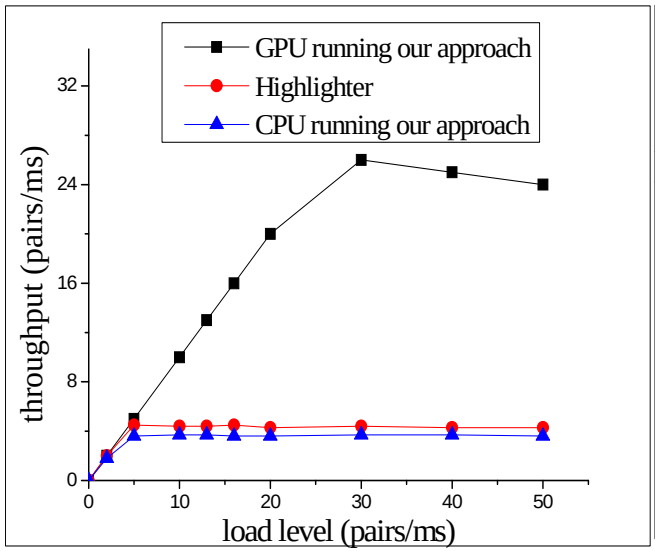
(b) Response time measurement



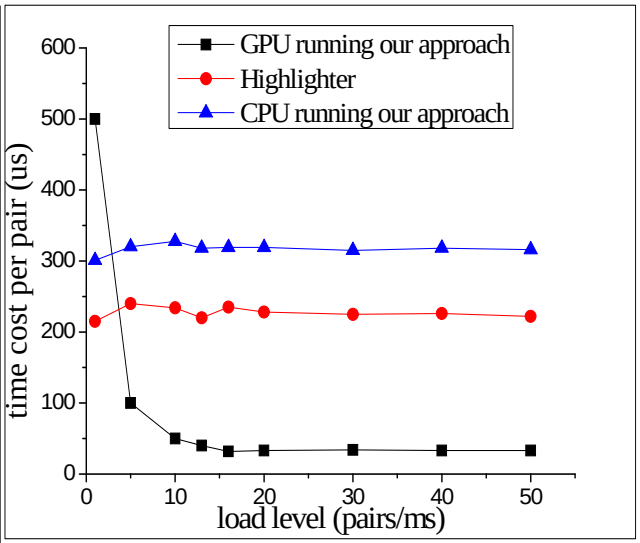
(c) Throughput measurement

Performance comparison tests

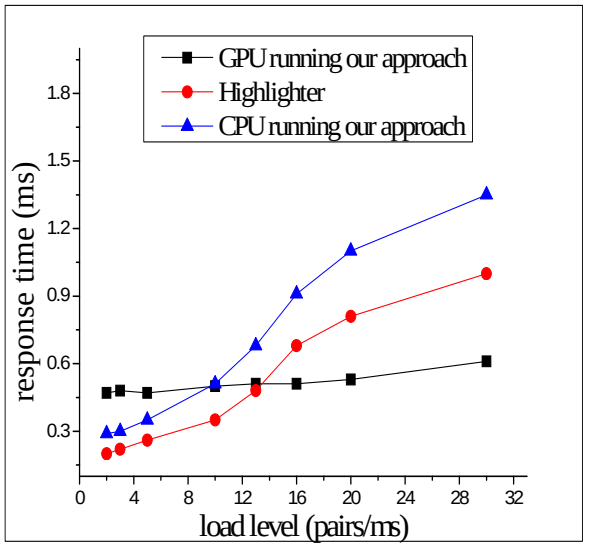
- The test results show that, when the work load is more than 5 pairs/ms, our approach has a much better performance than Highlighter in both throughput and average time cost by a pair.



(a) Throughput Measurement



(b) Time consuming of a pair



(c) Response time measurement

Economic advantage

- We compare our approach using GPU and CPU with other high performance multi-core processors.

	GTX250 + Intel Core2 Due(x2)	Intel Core2 Due (x2)	Intel i5 2300 (x4)	AMD Opteron8 6134 (x6)
Speed up	5.8	1.8	3.7	5.5
price($)	107+77=184	77	190	507

Snippet precision

- We compare our approach using GPU and CPU with Highlighter on CPU processor.
- We use harmonic mean of precision and recall (F1) as the precision measurement to test the snippet precision.

	r	p	F1
Our approach	0.48	0.44	0.459
Highlighter	0.46	0.43	0.444

Conclusions

- We propose a CPU-GPU hybrid system for snippet generation application of search engines.
- We employ sliding window for document segmentation.
- We parallelize the process of snippet generation for each segmentation and scoring.
- We design a pipeline system for stream processing that achieves a preferable performance.

Thanks!

http://idc.hust.edu.cn

Huazhong University of Science and Technology