Parallel Information Retrieval on A PC-Cluster Using Vector Space Model

Mr. Tawa Khampachua

Tawa.K@Student.chula.ac.th

Software Engineering Laboratory (SE)
Department of Computer Engineering
Chulalongkorn University
Problem Statement and Motivation

**Problem Statement**

- Large-scale text databases.
- Limitation of computing resources.

**Motivation**

- Emerging of parallel and distributed computing on Beowulf PC-Cluster.
Proposed Solution

Parallel computing
(On cluster of PCs)

+ Information Retrieval (using VSM model)

Parallel IR System

Large-scale Text Database
Information Retrieval

**What is Information Retrieval**

Information Retrieval is the process of identifying and retrieving relevant documents based on a query. It consists of three basic steps:

- A document representation.
- A query representation.
- A similarity measurement between both documents and queries.
The common classic models in information retrieval

- **Exact match**: based on text pattern and Boolean search techniques.
- **Inexact match**: based on statistic search techniques.
Vector Space IR Model

- Representing a document with a vector.
- Keeping document semantics with the length and direction of the vector.
**Retrieval Phase**

- Query vector formulation.
- Similarity measures; determining similarity of documents with simple vector operations (dot product).
Information Retrieval

IR system functional diagram

- Text DB
- Preprocess: stop words, stemming
- Indexing: vector computing
- Retrieval

- Stop words: eliminate non-useful words, e.g., a, an, the, etc.
- Stemming: reduce vocabulary size.
- Doc derive: construct document vectors.
- Retrieval: similarity computation.
Fundamental Theories - 6

Information Retrieval

Retrieval Evaluation

\[
\text{Recall} = \frac{Ra}{A} \\
\text{Precision} = \frac{Ra}{R}
\]
Parallel and Distributed Computing

Parallel computing concept:

# Decomposing a large problem into more parts.
# Processing each part in parallel by more processors.
# Reduce the time required to solve the problem.
# Scale up the larger problems.
Fundamental Theories - 8

Parallel and Distributed Computing

**SIMD and MIMD**

SIMD Architecture

MIMD Architecture
Performance Evaluation

- Speedup
  \[ S_p = \frac{T_1}{T_p} \]

- Efficiency
  \[ E_p = \frac{S_p}{p} \]
Problems and Solution

Problems

- A large-scale text database.
- A sequential system have a resources limitation.

Solution

- Apply parallel computing with IR system.
- Split a large text database into small pieces and conquer with computing nodes.
Parallel indexing using master-slave model
Parallel indexing functional diagram

- Preprocessing
  - Document Collection
  - Stemmed documents
- Split algorithm
  - Global df computation
  - Assign tid
  - df cutoff
- Local df computation
- Document vector computation
- Document vector weighting
- Document vector

Master
Slave
Parallel retrieval functional diagram

queries → query vectors Computation

Broadcast query vectors → Score Ranking

Score Ranking → Ranked documents list

query vectors → Similarity computation

document vectors →

Master

Slave
Experimental Results - 1

Materials and Configuration

**Collections**

- Test collection, Cranfield and TREC-8 (2GB).

**Testing**

- Retrieval performance (Precision).
- Speed-up performance.
## Experimental Results - 2

### Retrieval performance

#### Cranfiled Collections

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Size in Mbytes</td>
<td>1.6</td>
</tr>
<tr>
<td>Number of documents</td>
<td>1400</td>
</tr>
<tr>
<td>Number of queries</td>
<td>225</td>
</tr>
<tr>
<td>Maximum terms/document</td>
<td>225</td>
</tr>
<tr>
<td>Minimum terms/document</td>
<td>18</td>
</tr>
<tr>
<td>Average terms/document</td>
<td>88</td>
</tr>
<tr>
<td>Total terms found</td>
<td>125822</td>
</tr>
<tr>
<td>Total unique terms</td>
<td>3959</td>
</tr>
</tbody>
</table>
Experimental Results - 3

Retrieval performance

Test Diagram

Cranfield Collection

Stop words eliminate

Stemming

Stemmed documents

PVS
Indexing → Retrieval

SMART
Indexing → Retrieval

Evaluate

Software Engineering Laboratory
Experimental Results - 4

**Retrieval performance**

Retrieved Result

<table>
<thead>
<tr>
<th>Cranfiled Collection</th>
<th>PVS</th>
<th>SMART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>22500</td>
<td>22500</td>
</tr>
<tr>
<td>Relevant</td>
<td>1838</td>
<td>1838</td>
</tr>
<tr>
<td>Rel_retrieved</td>
<td>1295</td>
<td>1295</td>
</tr>
<tr>
<td>Average precision</td>
<td>0.3531</td>
<td>0.3531</td>
</tr>
</tbody>
</table>
**Experimental Results** - 5

**Speed-up performance**

**TREC Collections**

<table>
<thead>
<tr>
<th>Collections</th>
<th>LATIME</th>
<th>FR</th>
<th>FT</th>
<th>FBI S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total size (MB)</td>
<td>498</td>
<td>414</td>
<td>591</td>
<td>493</td>
</tr>
<tr>
<td>Total documents</td>
<td>131896</td>
<td>55630</td>
<td>210158</td>
<td>130471</td>
</tr>
<tr>
<td>Total words</td>
<td>70494552</td>
<td>38154925</td>
<td>84433831</td>
<td>67804697</td>
</tr>
<tr>
<td>Unique words</td>
<td>285643</td>
<td>157939</td>
<td>385795</td>
<td>321394</td>
</tr>
</tbody>
</table>
### Experimental Results - 6

**Speed-up performance**

#### Test Collections

<table>
<thead>
<tr>
<th>Computing nodes</th>
<th>Size of data (MB)</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,969</td>
<td>1,689,272</td>
</tr>
<tr>
<td>2</td>
<td>985</td>
<td>840,636</td>
</tr>
<tr>
<td>4</td>
<td>492</td>
<td>420,318</td>
</tr>
<tr>
<td>8</td>
<td>246</td>
<td>210,158</td>
</tr>
<tr>
<td>16</td>
<td>123</td>
<td>105,079</td>
</tr>
</tbody>
</table>
Experimental Results - 7

**Speed-up performance**

**Beowulf PCs cluster**

16 machines of Athlon 700MHz with 128MB RAM and IDE 10 GB.

Each of node is connected with 100Mbps Switching.
Experimental Results - 8

**Speedup curve**

- **Speed-up**
- **Computing nodes**

- Blue line: indexing
- Pink line: retrieval
Efficiency of indexing

Experimental Results - 9

Computing nodes

Speed-up

indexing
retrieval

Software Engineering Laboratory
Experimental Results - 10

Effect of Data Collection

Computing time

- FR
- FBIS
- FT

minutes

0 10 20 30 40 50 60 70 80 90
Experimental Results - 11

Comparison to SMART

Computing time

Software Engineering Laboratory
Experimental Results - 12

Comparison to SMART

Comparison of space requirements and maximum data size between PVS and SMART.

- Space:
  - PVS: 200 MB
  - SMART: 800 MB

- Maximum Data size:
  - PVS: 10 GB
  - SMART: 2 GB

Software Engineering Laboratory
Conclusion

Problem

• Large-scale text databases.
• Limitation of computing resources.

Propose

• Apply Parallel computing to information retrieval system.
• Using low-cost PCs cluster for implementation.

Consequence

• Parallel computing technique can help:
  • to reduce an execution time.
  • to solve resources limitation.
Conclusion

**Perspective**

- Apply Inverted file for improve speed of retrieval process.
- Optimize Parallel algorithms to reduce computing time.
- Add new functions and retrieval techniques to improve precision of system.