Parallelizing Information Retrieval

Gautam B. Singh
Department of Computer Science & Engineering
Oakland University, Rochester, MI

Parallelizing DNA Sequence DB Search

- Goal: Apply parallel algorithms for finding the neighbors of a query sequence in the database.
- Sequence Retrieval is one of the most common operations performed by biologists.
- GenBank (NIH) has 6 million records and some 5 billion characters.
Method and Activities

- **Method:** Use information theory based retrieval of sequences from database.
- **Current algorithms are based on finding Longest Common Subsequence (LCS).**
- **Data Set:** Data set from human chromosome sequences, about 2500 sequences.
- **Query:** 20 sequences of variable lengths from 500,000 to 2,000,000 characters in size.

Solution: Utilize a Distributed Parallel Paradigm

- Processors have distributed memory/database.
- An NxN Processor Interconnection Network is implemented by the WUGS switch.
Sequence Retrieval Algorithm

- Based on information theoretical measurements of sequence similarity.
- Constructs a profile or a signature of the sequences in the Database.
- Construct the profile of the query.
- Establish similarity by computing the distance between the two signatures.

Profile or Signature

- Profile = n-mer word frequency.
- An n-mer profile contains \((4^N)\) words.
- Consider the sequence: `ATACCGACCA`
- 4-mer profile (256 words): `ATAC, TACC, ACCA (2), CCAG, AGAC, GACC`.
- 6-mer profile (4096 words): `ATACCA, TACCAG, ACCAGA, CCAGAC, AGACCA`
Sequence Profile or Signature

Comparing Profiles

Profiles are normalized and converted into probability density functions.
Profiles are compared using the divergence measures.
**Retrieval System**

- Profiles for the sequences are pre-computed & stored.
- Retrieval system compares the query profile to all DB profiles.
- Neighbors are reported.

**Database and System Details**

- Our pilot database comprises of:
  - 2047 sequences
  - 500 MB data
- Two (2) Intel 133 Mhz CPUs connected via APIC/WUGS
- Profiles:
  - 8-mer profile (1.25 GB located on the centralized database)
Framework Possibilities

- Remote Method Interface
  - CORBA paradigm
  - The program is written with the INTERFACE
  - INTERFACE are implemented by RMI Server.
  - Programming model is really elegant-application parallelization is transparent.
  - This IS the parallel paradigm to program to!

Framework Possibilities (Cont.)

- MPI
  - Similar programming model as PVM
  - Still, have to explicitly parallelize write the parallel program
  - Do not have to work as hard as sockets

- Sockets
  - Explicit forking and synchronization
  - Difficult - Typically done after the parallel algorithm is well established.
### Parallel Application Frameworks

<table>
<thead>
<tr>
<th>Application Framework</th>
<th>Bandwidth</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVM: Parallel Virtual Machine</td>
<td>20 Mbps</td>
<td>Easy to program</td>
</tr>
<tr>
<td>MPI: Message Pass Interface</td>
<td>32 Mbps</td>
<td>Varies with block size.</td>
</tr>
<tr>
<td>Java RMI: Remote Method Invokation</td>
<td>32-36 Mbps</td>
<td>Ethernet and APIC performance comparable.</td>
</tr>
<tr>
<td>Sockets</td>
<td>100-200 Mbps</td>
<td>Varies. Best for packet size of 32 kb.</td>
</tr>
</tbody>
</table>

### Search Algorithm

- **8 Mer profiles for the DB to be searched is stored on the master processor**
- **Each query sequences communicated to all slaves.**
- **Slaves request work and get a database profile (0.5 MB message). Return the score and get new profile (work).**
Evaluating Parallel Performance

- Performance of parallelization was measured by Speedup:
  - Speedup = \( \frac{\text{Exec. Time 1 Proc.}}{\text{Exec. Time N Proc.}} \)
  - Our experiments, \( N = 2 \)
  - Ideal Speedup = 2

![8-Mer DB Search Speedup](chart.png)
Acknowledgements

- Washington University
- Ron Srodawa, Oakland University
- Donglin Liu, Oakland University