GBTK: A Toolkit for Grid Implementation of BLAST

Dr. Rajendra R. Joshi and Satish Kumar M.
rajendra@cdac.ernet.in
Coordinator, Bioinformatics
Scientific & Engineering Computing Group
C-DAC, Pune, India
http://bioinfo-portal.cdacindia.com
HIGH-THROUGHPUT TECHNIQUES ARE REVOLUTIONIZING LIFE SCIENCES

- DNA Sequencing
- Gene Expression Analysis With Microarrays
- Protein Profiling via High Throughput Mass Spectroscopy
- Protein-Protein Interactions
- Whole-Cell Response
Need of High Performance Computing in Bioinformatics

- Complete Published Genome Projects: 200
  - Archaeal: 19
  - Bacterial: 153
  - Eukaryal: 28
- Prokaryotic Ongoing Genome Projects: 508
- Eukaryotic Ongoing Genome Projects: 422
- http://www.genomesonline.org/

40.32 Gigabases from 35.53 million sequences
Release 142.0, June 2004
Computing For Life Sciences at the Terascale

Sequence Genome → Assemble → Gene Finding → “Identification” → Annotate → Gene to Protein → “Map” → Protein Interaction → Pathways Normal & Aberrant → Function in pathway → Drug Targets → Cellular Response

“Trivially Parallel” → “Massively Parallel”

Bioinformatics Molecular Biophysics Complex Systems
Grid Computing

- A type of parallel and distributed system that enables the sharing, selection and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost and users quality of service requirements.
GRID Initiatives in Life Sciences

- BioGRID  http://www.biogrid.jp
- NCBioGRID  http://www.ncbiogrid.jp
- APBioGRID  http://www.apbionet.org/apbiogrid/
- EuroGRID  http://www.eurogrid.org
- Canadian BioGRID  http://www.cbr.nrc.ca/
- MyGRID:  http://www.mygrid.org.uk
- TeraGrid:  http://www.teragrid.org
BLAST APPLICATION

- Basic Local Alignment Search Tool developed by Altschul et. al., in 1990


- Implements heuristic search method for finding maximal segment pairs (MSP) among a pair of sequences aligned

(1) For the query find the list of high scoring words of length \( w \).

Query Sequence of length \( L \)
- Maximum of \( L-w+1 \) words (typically \( w = 3 \) for proteins)

For each word from the query sequence find the list of words that will score at least 1 when scored using a parscore matrix (e.g., PAM 250). For typical parameters there are around 50 words per residue of the query.

(2) Compare the word list to the database and identify exact matches.

Database Sequences
- Exact matches of words from word list

(3) For each word match, extend alignment in both directions to find alignments that score greater than score threshold \( S \).

Maximal Segment Pairs (MSPs)
A list of words of size ‘W’ (e.g. W=4) are formed as an index of an array (an array of size $20^W$ for proteins)

For the **Query** find the list of high scoring words of length ‘W’. Compare the word list to the database and identify exact matches

For each word, extend alignment in both directions and find alignments that score greater than threshold score ‘S’
As BLAST algorithm is more selective and it can be best used for closely related sequences than for distantly related sequences. E.g. Similar sequences like ORFs, Paralogs, repeat elements etc.

BLAST programs are widely used for constructing Clusters of Orthologs (COGs) at NCBI (http://www.ncbi.nlm.nih.gov/COG).


BLAST is used at EMBL for finding orthologues (http://dove.embl-heidelberg.de/Blast2e/).

BLAST is also used in finding Alternate Splicing (AS) Sites.
Motivation

- To build a web based system that can be able to spawn BLAST jobs on heterogeneous PARAM supercomputers scattered across Indian cities of Bangalore/Pune.

Requirements:
- Needed an application specific Grid framework that will help to utilize distributed computing resources.
- Framework should be “simple” and should be able to work on machines of various configurations.
- A light weight framework, to spawn BLAST jobs intelligently and retrieve outputs.
Web Services

- Basis of GRID computing
- Services offered via the web
- Applications communicate and exchange data using XML RPC or SOAP
- Independent of underlying platform, operating system or programming language
XML-RPC

- What is XML-RPC?
  Remote Procedure Calling protocol with XML format

- What can it do?
  - allows software running on disparate operating systems, running in different environments to make procedure calls over the Internet.

- XML-RPC is composed by an HTTP request and a HTTP response.

- The body of the request and the value returned from server is formatted by XML.
XML-RPC
GBTK: Concept

- Virtualization
- Enabling seamless access
- Distributed data
- Connect geographically spread heterogeneous computing resources
- Portal interface for running BLAST jobs
Hardware Environment

- PARAM Padma cluster (AIX, 1 Teraflop, 248cpu)
- PARAM 10000 cluster (Solaris, 100 Gigaflop, 140cpu)
- PARAM OpenFrame (Solaris, 6 cpu)
- SGI Octane2 (IRIX, 2 cpu)
- Intel PIII (Linux, 1cpu)
- Intel PIII (Windows, 1 cpu)
Hardware Resources: PARAM PADMA

- Peak Computing Power - 1005 GF (~1 TF)
- Number of compute nodes - 54 Nos. of 4 Way SMP & 1 No. of 32 Way SMP
- No. of Processors - 248 (Power 4@1GHz)
- Aggregate Memory - 0.5 TeraBytes
- Internal Storage - 4.5 TeraBytes
- Operating System - AIX / LINUX
- Networks
  - PARAMNet-II @ 2.5 Gbps Full Duplex
  - Gigabit Ethernet @ 1 Gbps Full Duplex
- PARAMNet-II
  - in-house product
  - a high speed, low-latency switched network
  - Bandwidth - 2.5 Gbps
Hardware Resources: PARAM 10000

- Peak computing power of 100 Giga FLOPS
- Cluster of Sun Ultra e450 workstations 32 SMP compute nodes, each node with 4 processors (300 MHz)
- Physical memory: 1-2 GB
- Communication networks
  - Fast Ethernet
  - Myrinet
  - PARAMNet - in-house product
GRID BLAST

Domain Researchers

Web Browser

Computing Resources

PARAM 10000
GBTK: Features

- Application specific grid framework for BLAST
- Built on the concept of synchronized web services using RPC encoded as XML
- Light weight architecture
- Session tracking for distributed jobs
- Scheduling based on database availability and CPU load
- Capability of file transfer using remote copy protocol and secure copy protocol
Provides Portal based Interface & Hides Complexity to end user

Architecture

Steps:
- selectBestNode
- routeQuery
- encodeParameters2XML
- callRemoteBlast

Web Server
- getParameters Query, DB, Matrix
- getNodeStatus

Web services
- HTTP
- XML

Load Scheduling based on Size of Database and available Computing Resources.

Packed Request

Output

 Portal based Interface & Hides Complexity to end user
### Implementation: Database Distribution

Databases distributed across the computing nodes without redundancy.

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARAM Padma</td>
<td>PARAM 10000</td>
<td>PARAM OpenFrame</td>
<td>SGI Octane</td>
<td>Intel Box</td>
</tr>
<tr>
<td>OS: AIX</td>
<td>OS: Solaris</td>
<td>OS: Solaris</td>
<td>OS: IRIX</td>
<td>OS: Linux</td>
</tr>
<tr>
<td>EST_Human (2GB)</td>
<td>Swissprot (43MB)</td>
<td>PDB (3.82MB)</td>
<td>Prints (34MB)</td>
<td>Vector (3.7MB)</td>
</tr>
<tr>
<td>EST_Mouse (1GB)</td>
<td>Invertebrate (345MB)</td>
<td>Mitochondria (3.2MB)</td>
<td>Mammalian (31MB)</td>
<td>Syn P (0.9MB)</td>
</tr>
<tr>
<td>Viral (105MB)</td>
<td>Trembl (170MB)</td>
<td>E.coli (4.7MB)</td>
<td>Yeast (3.3MB)</td>
<td></td>
</tr>
<tr>
<td>Prokaryote (269MB)</td>
<td>NR (300MB)</td>
<td>Bacteriophage (4.9MB)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Implementation

- Web Services model consists of three components
  - Producer of web services
  - Broker which maintains the registry of available services
  - Consumer who consumes web services via the Broker
Implementation

- All computing nodes provide web services namely
  - CPU load
  - Application web service (BLAST)
  - Heart Beat
  - Initiate File Transfer
  - Receive File Transfer

- The Broker also provides a web service called DB Registry which contains locations of the databases.

- When the Broker gets a BLAST job request, with the aid of the DB registry it identifies the node on which the job should be executed.
**Scheduling**

- First Come First Serve model
- Based on:
  - Availability of target databases
  - CPU load

---

**Flowchart**

1. **Receive Job Request**
   - **Identify node where the Database is available**
     - **Is the machine free?**
       - Yes: **Route Job Request**
       - No: **Any other free nodes?**
         - No: **Queue**
         - Yes: **Route Job Request to newly identified node**
   - **Collect processed output**
User Interface

- GBTK provides a web based interface
- Uses CGI for receiving inputs from web pages
- Two categories
  - Master scripts: Retrieving inputs from the web and convert to XML & calling web services
  - Node scripts: Provide the web services functionality and wrappers for secured copy and remote copy data transfers
- Acknowledgment screen and status of job displayed
User Interface

- Web based Interface for the end user.
- Based on Apache Web server/ CGI.
Conclusion

- GBTK is based on Service Oriented Architecture
- Use of commodity tools will help in rapid deployment of application specific grids
- GBTK provides location transparency
- GBTK is a generic framework and can be used for any other application
Bioinformatics

- Molecular Modelling
- Metabolic Pathways
- Microarray Analysis
- Problem Solving Environments
- Protein Structure Prediction
- Ab-initio methods
- Genome Sequence Analysis

PARAM Padma
THANK YOU

contact: rajendra@cdacindia.com

http://bioinfo-portal.cdacindia.com