OmpiJava: Java Bindings for Open MPI

Dr.-Ing. Alexey Cheptsov
OmpiJava: Java Bindings for Open MPI

OUTLINE

- Motivation
  - Data-centric supercomputing
- Computing Platforms and Related Challenges
- Parallel Programming Models
- Java Bindings for OpenMPI
- Application Examples
  - Random Indexing
- What’s next?
Starting Points (1)

\[
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0
\]

\[
x\text{-momentum:} \quad \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right]
\]

\[
y\text{-momentum:} \quad \frac{\partial v}{\partial x} + u \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left[ \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right]
\]

- experimental
- theory
- computation
- data-driven

-1000
-100
-10

2013 Year, A.D.
Starting Points (2)

Evolution of Computational Applications

<table>
<thead>
<tr>
<th>Structure</th>
<th>Traditional Computational Sciences</th>
<th>Data-Intensive Sciences</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td></td>
<td>dynamic</td>
</tr>
</tbody>
</table>

| Locality         | spatially and temporally local       | no or little            |

| Volume           | fit into memory                     | do not fit into memory  |

| Arithmetical Complexity | High Precision Arithmetic | variable precision or integer based |

Teraflop WB 2013 :: 28.10.2013 :: Dr. Alexey Cheptsov
OmpiJava: Java Bindings for Open MPI

Data-centric Supercomputing

HPC in the “web data age”

- Why HPC is needed

Size of the data universe

<table>
<thead>
<tr>
<th>Year</th>
<th>Data Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>0.8 ZByte</td>
</tr>
<tr>
<td>2010</td>
<td>1.3 ZByte</td>
</tr>
<tr>
<td>2012</td>
<td>tens of ZBytes</td>
</tr>
</tbody>
</table>

Structured data ontologies linked data Semantic Web

<table>
<thead>
<tr>
<th>Year</th>
<th>2000</th>
<th>2005</th>
<th>2010</th>
<th>Year, A.D.</th>
</tr>
</thead>
</table>

Teraflop WB 2013 :: 28.10.2013 :: Dr. Alexey Cheptsov
Large-scale Graph Computing

- Google Knowledge Graph
  - 700 million nodes
  - 20 billion facts
  - several terabytes of files

- Facebook’s Social Graph
  - 60 PB of graph structured data

- Twitter’s Interest Graph

- NoSQL database solutions

Credit: Google,
http://www.stateofsearch.com/search-in-the-knowledge-graph-era/
OmpiJava: Java Bindings for Open MPI

Computing Platforms and Related Challenges

Semantic Web
Linked Data

Internet

Intranet

DATA-as-a-Service

Use of Supercomputers

Hermit – the HLRS mainstream system

- Cray XE6 architecture
- Performance of 1,2 PetaFLOP (10\(^{15}\) floating point operations per second)
- 3552 compute nodes
- 64GB RAM per node
- 2,7 PB disc space

Teraflop WB 2013 :: 28.10.2013 :: Dr. Alexey Cheptsov
What are the challenges

- Infrastructure „on demand“
  - distributed memory parallel clusters with low-latency intercon.
  - multicore machines with shared memory
  - GPGPU devices
  - alltogether?
OmpiJava: Java Bindings for Open MPI

What are the challenges

A Semantic Web Integration Platform for Large-Scale Reasoning

Teraflop WB 2013  ::  28.10.2013  ::  Dr. Alexey Cheptsov
LarKC architecture: High-Level Overview

Workflow branching

Plug-in parallelisation

Multi-Threading

MPI

Map-Reduce

Development Platforms

OmpiJava: Java Bindings for Open MPI

Computing Platforms and Related Challenges
What are the challenges

- Infrastructure „on demand“
  - shared/distributed memory parallel clusters
  - multicore machines
  - GPGPU devices
  - FPGA
  - altogether?
State-Of-The-Art: Global Multithreading over Shared Memory

- 128 TB Shared RAM
- 8192 HyperThreading CPU
- NTFS

YarcData’s uRiKA solution – a Jena-based framework on top of XMT

- non-standard pragma-driven instructions
- support of a limited set of languages
- high costs

**Use of the commodity HPC architectures will be the mainstream in the next 10 years of ICT research for data-intensive computing!**
What are the challenges

- Infrastructure „on demand“
  - shared/distributed memory parallel clusters
  - multicore machines
  - GPGPU devices
  - FPGA
  - altogether?

- Programming models to achieve high performance
  - MapReduce
  - MPI
  - „New“ programming languages

JUNIPER – Java platform for high performance and real-time large scale data management
State-Of-The-Art: Distributed Memory Processing over FS

- Programming models: MapReduce
  - data-centric
  - fault-tolerant
  - quite a poor performance
  - restrictive key/value model

Parallel Programming Models

OmpiJava: Java Bindings for Open MPI
The Message-Passing Interface

- Data-driven scenarios with MPI

- high performance

- integration ease
The Message-Passing Interface

Issues

- lack of implementations for the programming languages typically used in the data-centric communities, such as Java

• **Java implementations** (MPJExpress)
  - full MPI-2 standard implementation
  - issues with supporting new high-speed interconnects (e.g. Cray), related to the JVM
  - scalability to a peta/exaflop?
  - support of native tools for parallel computing, i.e. debuggers, error detectors, etc.

• **native implementations** (mpiJava)
  - JNI is used for calling communication libraries that are available in native codes (i.e. highly optimized MPI comm.)
  - integration with a native MPI library is not easy
  - …but if you got it running, very enjoyable performance
OmpiJava: Java Bindings for Open MPI

Parallel Programming Models

The Message-Passing Interface

➢ mpiJava

• Architecture

 Applications

 Java MPI bindings

 Native (C) MPI implementation

import mpi.*;

Java wrappers

JNI C Interface

Open MPI

MPICH
The Message-Passing Interface

- **Java bindings for Open MPI (ompiJava)**

  ![Diagram](image)

  - **User Application**
    - F77
    - F90
    - C
    - C++
    - JAVA
    - JNI
  - **OMPI**
  - **ORTE**
    - `mpirun`
  - **OPAL**
    - threads (OS), atomics (HW)
  - **OS** (Linux, Windows, MacOS)
  - Hardware (CPU, RAM, Network Interconnect)

  - **Configuration, installation**
    - `configure --enable-mpi-java`
    - `make install`
  - **Compilation**
    - `mpijavac Application.java`
  - **Running**
    - `mpirun java Application`
Developments @ HLRS

- ompiJava performance
  - P2P communication
Application Scenarios

- **Random Indexing for Large Texts**
  - A Statistical Distribution technique for word/text similarity analysis

![Diagram showing the process of random indexing for large texts](image)

**Terms** → **Occurrence vector** → **Docs**

**Subsetting**

**Query Expansion**

**SPARQL Query**

```
SELECT ?S ?P ?O
WHERE {
  FILTER (?O = 'ultrasound')
}
```

**Semantic Index**

- **Simularity**
  - 1.0
  - 0.96
  - 0.94
  - ...
- **Literal/URI**
  - ultrasound
  - reflection
  - sonography
  - ...

Teraflop WB 2013 :: 28.10.2013 :: Dr. Alexey Cheptsov
### Random Indexing: Pilot Data Sets (investigated by GATE)

<table>
<thead>
<tr>
<th>Data set</th>
<th>1 million (31GB)</th>
<th>36 million (400GB)</th>
<th>148 million (750GB)</th>
<th>289 million (2.88TB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#statements</td>
<td>150 million</td>
<td>2 billion</td>
<td>4 billion</td>
<td>23 billion</td>
</tr>
<tr>
<td>#terms</td>
<td>3133657</td>
<td>38 929 188</td>
<td>152 275 860</td>
<td>300 083 518</td>
</tr>
<tr>
<td>pre-processing</td>
<td>15 hours</td>
<td>28.3 hours</td>
<td>84 hours</td>
<td>160 hours</td>
</tr>
<tr>
<td>Generating Lucene index</td>
<td>23min (2.1G)</td>
<td>217min 36s (4G)</td>
<td>773min 21s (12G)</td>
<td>1777min 8s (42G)</td>
</tr>
<tr>
<td>Generating vectors (500, 10, 25)</td>
<td>9min (4.4G)</td>
<td>111min (13.4G)</td>
<td>459min 43s (50G)</td>
<td>996min 50s (84G)</td>
</tr>
<tr>
<td>search for the given word</td>
<td>53s</td>
<td>2min 43s</td>
<td>9min</td>
<td>14min 19s</td>
</tr>
</tbody>
</table>
Challenging Application Scenarios

- **Performance**
  - Pilot Application (Airhead Search)
Motivation

POI Analysis

I. Celino et. al. BOTTARI. Location based Social Media Analysis with Semantic Web. SWC 2011.
OmpiJava: Java Bindings for Open MPI

Some additional features

- **omniaJava profiling architecture**
- MPITrace (Extrae)
- VampirTrace

Profiling Interface (PMPI)

Applications

Java wrappers

JNI C Interface

Native (C) MPI implementation

Open MPI

MPICH

Teraflop WB 2013 :: 28.10.2013 :: Dr. Alexey Cheptsov
Extrae (MPITrace)

- MPI profiling library
- Uses late dynamic binding of the profiling code (with Dyninst, linker preload)

```bash
export LD_PRELOAD=${EXTRAE_HOME}/lib/libmpitrace.so
export EXTRAENAME=1
export EXTRAENAME_CONFIG_FILE=./extrae/extrae.xml

<set enabled="yes" domain="all" changeatglobalops="100">
  PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_L2_DCM,
  PAPI_TLB_DM,PAPI_FP_INS
</set>
```
Trace Visualization with Paraver

- Tool for visualization trace files generated with Extrae

<table>
<thead>
<tr>
<th>Thread</th>
<th>Running</th>
<th>Not created</th>
<th>Waiting a message</th>
<th>Blocked</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1.1.1</td>
<td>19,367,517,071 ns</td>
<td>56,506,575 ns</td>
<td>52,289,921 ns</td>
<td>56,506,575 ns</td>
</tr>
<tr>
<td>T1.2.1</td>
<td>15,842,887,731 ns</td>
<td>42,517,272 ns</td>
<td>64,145,871 ns</td>
<td>42,517,272 ns</td>
</tr>
<tr>
<td>T1.3.1</td>
<td>15,715,187,818 ns</td>
<td>50,276,257 ns</td>
<td>84,570,364 ns</td>
<td>50,276,257 ns</td>
</tr>
<tr>
<td>T1.4.1</td>
<td>10,276,215,224 ns</td>
<td>14,770 ns</td>
<td>42,516,284 ns</td>
<td>14,770 ns</td>
</tr>
</tbody>
</table>

- Total: 61,201,807,844 ns
- Average: 15,300,451,961 ns
- Maximum: 19,367,517,071 ns
- Minimum: 10,276,215,224 ns
- StDev: 3,250,008,299.76 ns
- Avg/Max: 0.79/0.66

- Step 1. Broadcasting base vector
  - MPI_Send/Recv, MPI_Broadcast

- Step 2 (parallel). Vector processing

- Step 3. Gathering results
  - MPI_Gather

Teraflop WB 2013 :: 28.10.2013 :: Dr. Alexey Cheptsov
Main Results

- HPC is going to face new challenges related to data-centric application expansion.
- Parallel programming models (mainly MapReduce and MPI) are the key enablers of HPC to data-centric applications.
- Reaching near-peak performance is going to be the major challenge.

Future Work

- Promote existing technologies, such as MPI, to solving new challenges, such as Big Data.
- Making existing framework more data-centric.