Charm++ - Feature #1666
Support # 1586 (New): OpenMP integration support for external library written in and compiled by OpenMP on Charm++

Standalone OpenMP implementation based on Converse User-level threads
08/25/2017 01:34 PM - Seonmyeong Bak

<table>
<thead>
<tr>
<th>Status:</th>
<th>New</th>
<th>Start date:</th>
<th>09/06/2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>Seonmyeong Bak</td>
<td>% Done:</td>
<td>0%</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
</tbody>
</table>

Description
The current OpenMP implementation only works with Charm++ and AMPI. However, this OpenMP implementation basically can work standalone because it only uses converse APIs which are independent of Charm++ and AMPI.

Several things should be done for this but it doesn't require huge changes in the current implementation.

1. Move the task queue into converse level
   The current task queue is used only for OpenMP and there's no reason that it should be in ck-core. It only uses converse constructs.

2. Initialization of Converse library in OpenMP runtime.
   LLVM OpenMP runtime has explicit initialization point so we only need to call 'ConverseInit' there and 'CsdExitScheduler' in the termination of the scheduler.

This work can bring many benefits other than the modularity of the OpenMP implementation.

1. We can accelerate all existing OpenMP applications with our work.
   Because the OpenMP works on converse and can use all the feature of the converse, we can provide better OpenMP implementation compared to the vanilla LLVM OpenMP and other OpenMP implementations including research OpenMP implementations such as ForestGOMP and BOLT.
   In current implementation, we can accelerate only MPI + OpenMP applications other than Charm++ applications. However, MPI and Charm++ are only used for scientific applications, which have very limited pool of users.
   If we want to see some applications we haven't touched, we should be able to run other parallel apps beyond MPI and Charm++.
   OpenMP has bigger user pool and been used beyond academia and scientific applications such as machine learning and data analysis apps directly or indirectly through libraries.

2. OpenMP with other high-level languages and parallel frameworks.
   There are many parallel frameworks in high-level languages which adopts fork-join model instead of message passing model.
   We can accelerate these frameworks with our OpenMP in many ways. Even though we're working on python support of Charm++, it just ports Charm++ constructs into python and still highly complex to use for ordinary users.

Subtasks:
Feature # 1670: Migration of Taskqueue from Ck to Converse layer

History
#1 - 08/25/2017 01:34 PM - Seonmyeong Bak
  - Assignee set to Seonmyeong Bak

#2 - 09/05/2017 11:56 PM - Seonmyeong Bak
  - % Done changed from 0 to 50

Basically, it works with a few lines of codes in converse and OpenMP runtime.