Charm++ - Feature #1953

cmake-based build system

07/31/2018 02:14 PM - Matthias Diener

<table>
<thead>
<tr>
<th>Status:</th>
<th>In Progress</th>
<th>Start date:</th>
<th>07/31/2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>Matthias Diener</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>6.9.1</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
</tbody>
</table>

Description

Our current home-grown build system consists of:

- the ./build shell script
- the ./smart-build.pl Perl script
- an autoconfiscated configure script
- an assortment of hand-written Makefiles
- completely separate build systems (hwloc, ROMIO, ...)

It would be interesting to have a more standard approach to building Charm++:

- Users are more familiar with the build system.
- We can provide reasonable defaults, such that users do not have to specify the network layer, SMP mode, optimization options etc.
- Fix issues with the current build system (e.g., Makefiles that ignore errors, infinite recompilation of some files, ...)

CMake is probably the most interesting alternative for us:

- It is supported on all platforms we support.
- It is installed on most systems.
- It only requires a C++ compiler.
- It has integrated testing and packaging frameworks.

Related issues:

Related to Charm++ - Feature #1098: Better CMake support  Merged  06/10/2016

History

#1 - 08/03/2018 04:38 PM - Sam White
https://charm.cs.illinois.edu/gerrit/#/c/charm/+/4418/

#2 - 08/12/2018 11:51 AM - Matthias Diener
- Assignee set to Matthias Diener
- Status changed from New to In Progress

#3 - 08/17/2018 05:34 PM - Evan Ramos
- Related to Feature #1098: Better CMak support added

#4 - 09/06/2018 02:23 PM - Evan Ramos
- Target version set to 6.9.1