Leveraging the checkpoint-restart technique for optimizing CPU efficiency of ATLAS production applications on opportunistic platforms

D Cameron, J Elmsheuser, L Heinrich, W Lavrijsen, P Nilsson, V Tsulaia and M Vogel
for the ATLAS Collaboration
Introduction

- ATLAS uses **AthenaMP** (multi-process version of its data reconstruction, simulation and analysis framework Athena) for running production workloads on multi-core platforms.

- The sequential phase of an Athena MP job (initialization in the master process) includes:
  - Loading of shared libraries
  - Reading of the detector geometry and conditions data from external databases
  - Building transient representation of the detector geometry
  - Initialization of algorithms and services

- In some cases the **initialization step takes as long as 10-15 minutes**, which has a visible negative impact on overall CPU efficiency of the job.
ATLAS runs large number of production jobs with similar configuration parameters

- E.g. jobs within one Geant4 Simulation production task share the same configuration
- The only difference between jobs within such task is which events they need to process

For such jobs, instead of going through the same initialization phase over and over again, we can consider the following scenario:

- **Step 1:** run one job from the task through the initialization step and then checkpoint it;
- **Step 2:** distribute the generated **checkpoint image** over to the compute nodes;
- **Step 3:** restart production jobs from the checkpoint image instead of running them through the initialization phase.

This presentation contains some preliminary results of applying this strategy to ATLAS Geant4 Simulation jobs on a Volunteer Computing system and an Intel KNL supercomputer.
Checkpointing Tool

- **DMTCP** (Distributed MultiThreaded Checkpointing) [http://dmtcp.sourceforge.net/](http://dmtcp.sourceforge.net/)
  - Checkpoints a single host or distributed computation in user space
  - Does not require kernel-level access
  - Minimum runtime overhead

- Has been used for testing the checkpoint-restart mechanism for CMSSW and Geant4 MT
  - See the presentation by P Elmer in ACAT 2013

- For our tests so far we have been using DMTCP 2.4.5
  - Integrated into ATLAS software releases as an external package
Making checkpoint images in AthenaMP

❖ The **master AthenaMP process checkpoints itself** just before forking event processors
  ❖ Using DMTCP API from within C++ code

❖ Once the checkpoint image has been created, the master process exits immediately

❖ The checkpoint image together with other auxiliary files (e.g. automatically generated script for restarting) is put into a tarball for later usage

❖ In order to activate this mechanism we introduced a special --checkpoint command-line option to ATLAS Job Transform
  ❖ **Job Transform** is a python wrapper used for running Athena jobs in production
Restarting AthenaMP from a checkpoint image

❖ The location of a checkpoint image is provided to the transform via --restart command-line option

❖ The transform unpacks the checkpoint tarball into job’s run directory and initiates restart by running the restart script

❖ The first thing AthenaMP does after restart is to update a few configuration parameters: numbers of processes to fork, input file name and the number of events to process
  ❖ This information is provided to AthenaMP by the transform

❖ After that the job proceeds as usual
Portability of checkpoint images

- It is desirable to generate one checkpoint image for a large set of jobs (e.g., all jobs within the same production task) and then use it for launching jobs on heterogeneous platforms at various sites.
  - ATLAS software is built against SLC6 for production usage.
- This is non-trivial, for DMTCP expects to see the same platform at restart as the one seen at checkpoint.
- In order to overcome this limitation, we can leverage VM/container technology.
  - Create checkpoint image within VM/container and restart within the same VM/container.
- We followed this strategy for testing AthenaMP checkpoint-restart on BOINC.
  - Volunteer computing platform used by the ATLAS@Home project.
Testing on ATLAS@Home

❖ ATLAS@Home
  ❖ A volunteer computing project started in 2014
  ❖ An outreach tool to get the public involved in ATLAS
  ❖ Volunteers run Geant4 simulation inside a VM
  ❖ Based on BOINC platform used in many volunteer computing projects
  ❖ Built-in support for virtualization using VirtualBox
  ❖ ATLAS@Home is integrated into the ATLAS workflow management system
  ❖ From the outside it looks like a normal “Grid” site
  ❖ ATLAS@Home is currently an equivalent of a T2 site

Events processed on ATLAS@Home per month since the start of the project. The colors show the transition from single-core (red) to multi-core (yellow) jobs

V Tsulaia et al, ATLAS, ACAT 2017
Testing on ATLAS@Home (contd.)

- ATLAS@Home makes a good choice for a prototype platform to test checkpoint-restart
  - Since jobs run in a VM, we are in a complete control of the environment
  - Volunteers often complain of long initialization times due to reading of external databases
  - Jobs in ATLAS@Home/BOINC run for 1-2hr, which makes fast initialization rather important

- Volunteers download a VM disk image once and use it for each job
  - We create an image with all the necessary software cached in the CVMFS cache

- For checkpointing tests we created a tarball with a compressed checkpoint image and saved it in the VM image

- Preliminary tests demonstrated that AthenaMP can restart from the checkpoint image in 15-20 sec, while in case of regular initializations we measured ~4 min for jobs with fast database connections and 10-15 min for jobs with slow database connection
Cori HPC @ NERSC

- The 5th most powerful supercomputer in the world on the November 2016 list of Top 500 HPCs
- Cori Phase 1: 2.4K Intel Xeon “Haswell” nodes
- Cori Phase 2: 9.7K Intel KNL nodes
  - 68 cores per node @ 1.4 GHz
  - Each core has 4 hardware threads and two 512-bit-wide vector processing units
  - Each node has 96 GB DDR4 + 16 GB MCDRAM (multi-channel DRAM) memory
- In July 2017 ATLAS simulated 55M events with Geant4 on Cori Phase 1&2.
Testing on Intel KNL (contd.)

- On Cori KNL we tested AthenaMP restarts from locally generated compressed and uncompressed images.
- Cori compute nodes represent a homogenous environment, so we don’t have to worry about image portability.
- The image tarballs were unpacked into run directories prior to submitting jobs to the batch system.
- The job startup time was measured between launching AthenaMP until the moment it forked event processors.
- The table contains results obtained by running 300 singe-node jobs.

<table>
<thead>
<tr>
<th>Image Size</th>
<th>Startup Time (sec)</th>
<th>Startup Speedup vs Regular AthenaMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular AthenaMP</td>
<td>N/A 663.1 ± 22.8</td>
<td>1</td>
</tr>
<tr>
<td>Compressed image</td>
<td>550MB 50 ± 9.7</td>
<td>13.3x</td>
</tr>
<tr>
<td>Uncompressed image</td>
<td>1.8GB 20.8 ± 9.1</td>
<td>31.5x</td>
</tr>
</tbody>
</table>
DMTCP was successfully tested by ATLAS for checkpoint-restarting Geant4 Simulation jobs.

First tests with production jobs on ATLAS@Home/BOINC and Intel KNL demonstrated that by restarting from checkpoint images we can considerably speedup job startup times.

However, there is still a long way to go before we can declare our readiness to use the checkpoint-restart technology in production.

- Automation of the process
- Validation of the results

The authors would like to thank our volunteer testers in ATLAS@Home (Yeti and MAGIC) for running the jobs for us and providing useful feedback.