UPDATE ON ATLAS SOFTWARE BUILDS

Johannes Elmsheuser
9 March 2022, ATLAS Google meeting
ATLAS SOFTWARE SCHEMATIC OVERVIEW

Projects

- Athena
  - 1933 packages
- AthDerivation/21.2
  - 922 packages
- AthSimulation
  - 370 packages
- AthAnalysis
  - 294 packages
- AthGeneration
  - 230 packages
- DetCommon
  - 15 packages
- AnalysisBase
  - 199 packages

Externals

- AthenaExternals
  - ~200 packages
  - ROOT, Gaudi, Geant4, Python, Generators, ...
- Parts compiled in build
- Parts provided by LCG
- tdaq+tdaq-common
  - ~200 packages
  - only need for some Projects provides COOL/CORAL

Compiled with gcc11, gcc8, clang10, clang13 on CentOS7 for x86_64
AthSimulation for aarch64
A full ATLAS SW framework Athena nightly build starts in the evening CERN time and can take up to 10h until SW is available on CVMFS for direct world-wide access - subsequently many integration tests are started on the Grid (via the ART system on PanDA) which adds several hours in addition

- Every gitlab merge request (MR) runs a CI pipeline with incremental code builds, ctests and some local integration tests - takes 2-5h per MR

→ Shorter build times reduce the wait time for developers

- In addition: explore new architectures and compare Cloud infrastructure to existing CERN nodes
· **aibuild16-042.cern.ch:**
  · Intel Xeon E5-2683 v4 (Broadwell), April 2016, 16 CPU cores, 120 GB RAM

· **techlab-arm64-thunderx2-01.cern.ch:**
  · Cavium ThunderX2 99xx, 256 CPU cores, 256 GB RAM

· **AmazonWebServices:** (larger/smaller EC2 setups configurable)
  · ARM Neoverse Graviton, chose c6gd.4xlarge, 16 CPU Cores, 32 GB RAM, 300 GB EBS storage
  · Added CVMFS, no HepOSLib meta RPM package for ARM, manual installation

· **Google:** (larger/smaller GCP setups configurable)
  · HPC VM image: c2-standard-30, 30 vCPU (Intel Cascade Lake), April 2019 or newer, 120 GB RAM, 250 GB SSD persistent disk
  · VM image 2: c2d-standard-32/112, 32/122 vCPU (AMD EPYC Milan), March 2021, 128/448 GB RAM, 250 GB SSD persistent disk
  · HPC VM image: c2-standard-16, 16 vCPU (Intel Cascade Lake), 64 GB RAM, 500 GB Extreme persistent disk
  · CentOS 7.9, added CVMFS (25 GB), HepOSLib meta RPM package, openssl
  · No EOS fuse-mount to CERN - some ctests use reference files from EOS
### Software Build Time Comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aibuild20-37</td>
<td>20</td>
<td>60</td>
<td>AthSimulation</td>
<td>x86_64</td>
<td>gcc11</td>
<td>RelWithDebInfo</td>
<td>26</td>
<td>4.2</td>
<td>39</td>
<td>7</td>
</tr>
<tr>
<td>thunderx2</td>
<td>256</td>
<td>256</td>
<td>AthSimulation</td>
<td>aarch64</td>
<td>gcc8</td>
<td>RelWithDebInfo</td>
<td>15</td>
<td>5.8</td>
<td>27.7</td>
<td>13</td>
</tr>
<tr>
<td>AWS</td>
<td>16</td>
<td>32</td>
<td>AthSimulation</td>
<td>aarch64</td>
<td>gcc8</td>
<td>RelWithDebInfo</td>
<td>18</td>
<td>3.5</td>
<td>30.3</td>
<td></td>
</tr>
<tr>
<td>aibuild16-042</td>
<td>16</td>
<td>120</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>RelWithDebInfo</td>
<td>28</td>
<td>8.7</td>
<td>347</td>
<td>103</td>
</tr>
<tr>
<td>aibuild16-025</td>
<td>16</td>
<td>120</td>
<td>Athena</td>
<td>x86_64</td>
<td>clang10</td>
<td>RelWithDebInfo</td>
<td>42</td>
<td>6.7</td>
<td>274</td>
<td></td>
</tr>
<tr>
<td>aibuild16-027</td>
<td>16</td>
<td>120</td>
<td>Athena</td>
<td>x86_64</td>
<td>clang13</td>
<td>RelWithDebInfo</td>
<td>28</td>
<td>4.5</td>
<td>277</td>
<td>-</td>
</tr>
<tr>
<td>AWS</td>
<td>16</td>
<td>32</td>
<td>Athena*</td>
<td>aarch64</td>
<td>gcc8</td>
<td>Release</td>
<td>15.7</td>
<td>3.5</td>
<td>194</td>
<td></td>
</tr>
<tr>
<td>AWS</td>
<td>16</td>
<td>32</td>
<td>Athena*</td>
<td>aarch64</td>
<td>gcc8</td>
<td>RelWithDebInfo</td>
<td>19.5</td>
<td>4.3</td>
<td>258</td>
<td></td>
</tr>
<tr>
<td>AWS</td>
<td>16</td>
<td>32</td>
<td>Athena</td>
<td>aarch64</td>
<td>gcc11</td>
<td>Release</td>
<td>18.4</td>
<td>4.3</td>
<td>206</td>
<td></td>
</tr>
<tr>
<td>Google (c2, SSD)</td>
<td>30</td>
<td>120</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>Release</td>
<td>15.4</td>
<td>4.2</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>Google (c2, SSD)</td>
<td>30</td>
<td>120</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>RelWithDebInfo</td>
<td>16.7</td>
<td>3.1</td>
<td>182</td>
<td></td>
</tr>
<tr>
<td>Google (c2, SSD)</td>
<td>30</td>
<td>120</td>
<td>Athena</td>
<td>x86_64</td>
<td>clang13</td>
<td>RelWithDebInfo</td>
<td>19.0</td>
<td>3.2</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>Google (c2d, SSD)</td>
<td>32</td>
<td>128</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>Release</td>
<td>15.0</td>
<td>3.7</td>
<td>172</td>
<td></td>
</tr>
<tr>
<td>Google (c2, EPD)</td>
<td>16</td>
<td>64</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>Release</td>
<td>24.4</td>
<td>3.9</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>Google (c2, EPD)</td>
<td>16</td>
<td>64</td>
<td>Athena</td>
<td>x86_64</td>
<td>clang13</td>
<td>Release</td>
<td>23.2</td>
<td>5.0</td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>Google (c2, local SSD)</td>
<td>16</td>
<td>64</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>Release</td>
<td>25.7</td>
<td>7.3</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>Google (c2, local SSD)</td>
<td>16</td>
<td>64</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>RelWithDebInfo</td>
<td>28.7</td>
<td>3.4</td>
<td>319</td>
<td></td>
</tr>
<tr>
<td>Google (c2, SSD)</td>
<td>60</td>
<td>240</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>Release</td>
<td>12.1</td>
<td>4.2</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>Google (c2, SSD)</td>
<td>60</td>
<td>240</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>RelWithDebInfo</td>
<td>12.3</td>
<td>3.5</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>Google (c2, SSD)</td>
<td>60</td>
<td>240</td>
<td>Athena</td>
<td>x86_64</td>
<td>clang13</td>
<td>Release</td>
<td>11.7</td>
<td>3.5</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>Google (c2, SSD)</td>
<td>30</td>
<td>120</td>
<td>Athena</td>
<td>x86_64</td>
<td>clang13</td>
<td>Release</td>
<td>18.4</td>
<td>4.1</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>Google (c2d, SSD)</td>
<td>112</td>
<td>448</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>Release</td>
<td>8.5</td>
<td>4.2</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>Google (c2d, SSD)</td>
<td>112</td>
<td>448</td>
<td>Athena</td>
<td>x86_64</td>
<td>gcc11</td>
<td>RelWithDebInfo</td>
<td>8.7</td>
<td>3.4</td>
<td>132</td>
<td></td>
</tr>
<tr>
<td>Google (c2d, SSD)</td>
<td>112</td>
<td>448</td>
<td>Athena</td>
<td>x86_64</td>
<td>clang13</td>
<td>Release</td>
<td>10.6</td>
<td>3.5</td>
<td>77</td>
<td></td>
</tr>
</tbody>
</table>

- Google nodes look good - optimization of CVMFS and local disks for better I/O required
- CERN nodes could benefit from CPU hardware update and I/O improvements (?)
### Athena Build Time Comparison vs. CPUs on Google

- **Google**
  - 250 GB SSD persistent disk: shares local code and 25 GB CVMFS cache - not sure if setup is optimal
  - c2-standard-x, for 16, 30, 60 vCPUs (Intel Cascade Lake, April 2019 or newer), 64/120/240 GB RAM
  - c2d-standard-112, 112 vCPUs (AMD EPYC Milan, March 2021), 448 GB RAM
- **CERN**
  - aibuild16-042, 16 vCPUs, 120 GB RAM
GOOGLE: C2D (112 vCPUs), CPU and I/O for 3 builds

- First 20-30min of the build are used for inefficient (=not highly parallel) dictionary and python/shell file creation
- I/O: 2-5 MiB/s write for Release, and 20-30 MiB/s for RelWithDebInfo during compilation
- 112 vCPUs seem not to be fully utilised on average (compare also to 60 vCPU build time) or hitting some I/O bottlenecks - need to try local SSD RAID
First 20-30min of the build are used for inefficient (=not highly parallel) dictionary (genreflex) and python/shell file creation

Note rather high virtual memory of cvmfs process (pre se not a problem)