Interoperability via common Build & Test (BaT)

Miron Livny
Computer Sciences Department
University of Wisconsin-Madison
Thesis

Interoperability of middleware can only be achieved if all components can be built and tested in a common Build & Test (BaT) infrastructure

- Necessary but not sufficient
- Infrastructure must be production quality and distributed
- Software must be portable
- A community effort that leverages know-how and software tools
Motivation

› Experience with the Condor software
  • Includes external dependencies and interacts with external middleware
  • Ported to a wide range of platforms and operating systems
  • Increasing demand for automated testing

› Experience with the Condor community
  • How Oracle has been using Condor for their build and test activities
  • Demand from “power users” for local BaT capabilities
The NSF Middleware Initiative (NMI) Build and Test Effort
GRIDS Center
- Enabling Collaborative Science-

Grid Research Integration Development & Support
The NMI program

- Program launched by Alan Blatecky in FY02
- ~ $10M per year
- 6 “System Integrator” Teams
  - GRIDS Center
    - Architecture and Integration (ISI)
    - Deployment and Support (NCSA)
    - Testing (UWisc)
  - Grid Portals (TACC, UMich, NCSA, Indiana, UIC)
  - Instrument Middleware Architecture (Indiana)
  - NMI-EDIT (EDUCAUSE, Internet2, SURA)
- 24 Smaller awards developing new capabilities
NMI Statement

• Purpose - to develop, deploy and sustain a set of reusable and expandable middleware functions that benefit many science and engineering applications in a networked environment

• Program encourages open source software development and development of middleware standards
The Build Challenge

› Automation - “build the component at the push of a button!”
  • always more to it than just “configure” & “make”
  • e.g., ssh to right host; cvs checkout; untar; setenv, etc.
› Reproducibility - “build the version we released 2 years ago!”
  • Well-managed & comprehensive source repository
  • Know your “externals” and keep them around
› Portability - “build the component on nodeX.cluster.com!”
  • No dependencies on “local” capabilities
  • Understand your hardware & software requirements
› Manageability - “run the build daily on 15 platforms and email me the outcome!”
The Testing Challenge

All the same challenges as builds (automation, reproducibility, portability, manageability), plus:

Flexibility

- “test our RHEL4 binaries on RHEL5!”
- “run our new tests on our old binaries”
- important to decouple build & test functions
- making tests just a part of a build -- instead of an independent step -- makes it difficult/impossible to:
  - run new tests against old builds
  - test one platform’s binaries on another platform
  - run different tests at different frequencies
Depending on our own software

▷ What Did We Do?
  • We built the NMI Build & Test facility \textit{on top of} Condor, Globus and other distributed computing technologies to automate the build, deploy, and test cycle.
  • To support it, we’ve had to construct and manage a dedicated, heterogeneous distributed computing facility.
    • Opposite extreme from typical “cluster” -- instead of 1000’s of identical CPUs, we have a handful of CPUs each for \( \sim 40 \) platforms.
    • Much harder to manage! You try finding a sysadmin tool that works on 40 platforms!

▷ We’re just another demanding grid user - If the middleware does not deliver, we feel the pain!!
## Numbers

<table>
<thead>
<tr>
<th>Name</th>
<th>Arch</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>/atlantis.mcs.anl.gov</td>
<td>sparc</td>
<td>sol9</td>
</tr>
<tr>
<td>/grandcentral</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/janet</td>
<td>i386</td>
<td>winxp</td>
</tr>
<tr>
<td>/nmi-build15</td>
<td>i386</td>
<td>rh72</td>
</tr>
<tr>
<td>/nmi-build16</td>
<td>i386</td>
<td>rh8</td>
</tr>
<tr>
<td>/nmi-build17</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-build18</td>
<td>sparc</td>
<td>sol9</td>
</tr>
<tr>
<td>/nmi-build21</td>
<td>i386</td>
<td>fc2</td>
</tr>
<tr>
<td>/nmi-build29</td>
<td>sparc</td>
<td>sol8</td>
</tr>
<tr>
<td>/nmi-build33</td>
<td>ia64</td>
<td>sl6es</td>
</tr>
<tr>
<td>/nmi-build5</td>
<td>i386</td>
<td>rhel3</td>
</tr>
<tr>
<td>/nmi-build6</td>
<td>i386</td>
<td>osx</td>
</tr>
<tr>
<td>/nmi-rhas3-amd64</td>
<td>amd64</td>
<td>rhel3</td>
</tr>
<tr>
<td>/nmi-sles8-amd64</td>
<td>amd64</td>
<td>sl6es</td>
</tr>
<tr>
<td>/nmi-test-3</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-4</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-9</td>
<td>hp</td>
<td>hpux11</td>
</tr>
<tr>
<td>/nmi-test-10</td>
<td>sgi</td>
<td>irix67</td>
</tr>
<tr>
<td>/nmi-test-11</td>
<td>sparc</td>
<td>sol10</td>
</tr>
<tr>
<td>/nmi-test-12</td>
<td>sparc</td>
<td>sol7</td>
</tr>
<tr>
<td>/nmi-test-13</td>
<td>sparc</td>
<td>sol8</td>
</tr>
<tr>
<td>/nmi-test-14</td>
<td>sparc</td>
<td>sol9</td>
</tr>
<tr>
<td>/nmi-test-15</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-16</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-17</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-18</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-19</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-20</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-21</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-22</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-23</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-24</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-25</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-26</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-27</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-28</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-29</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-30</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-31</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-32</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-33</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-34</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-35</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-36</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-37</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-38</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-39</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-40</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-41</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-42</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-43</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-44</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-45</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-46</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-47</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-48</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-49</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-50</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-51</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-52</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-53</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-54</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-55</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-56</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-57</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-58</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-59</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-60</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-61</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-62</td>
<td>i386</td>
<td>rh9</td>
</tr>
<tr>
<td>/nmi-test-63</td>
<td>i386</td>
<td>rh9</td>
</tr>
</tbody>
</table>

- **100 CPUs**
- **39 HW/OS “Platforms”**
- **34 OS**
- **9 HW Arch**
- **3 Sites**
- **~100 GB of results per day**
- **~1400 Builds/tests per month**
- **~350 Condor jobs per day**
Condor Build & Test

› Automated Condor Builds
  • Two (sometimes three) separate Condor versions, each automatically built using NMI on 13-17 platforms nightly
    • Stable, developer, special release branches

› Automated Condor Tests
  • Each nightly build’s output becomes the input to a new NMI run of our full Condor test suite

› Ad-Hoc Builds & Tests
  • Each Condor developer can use NMI to submit ad-hoc builds & tests of their experimental workspaces or CVS branches to any or all platforms
### Condor Latest Build/Test Results

<table>
<thead>
<tr>
<th>Branch</th>
<th>Last build</th>
<th>User</th>
<th>Build Results</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>V6.6-branch</td>
<td>2006-04-25 08:05:04</td>
<td>cndrauto</td>
<td>PENDING</td>
<td>PENDING</td>
</tr>
<tr>
<td>V6.7-branch</td>
<td>2006-04-25 08:05:18</td>
<td>cndrauto</td>
<td>PENDING</td>
<td>PENDING</td>
</tr>
<tr>
<td><strong>One-off Builds</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V6.6-branch</td>
<td>2006-04-25 03:23:03</td>
<td>akarp</td>
<td>PASSED</td>
<td>PASSED</td>
</tr>
<tr>
<td>V6.7-Technion_has-branch</td>
<td>2006-04-25 04:50:14</td>
<td>dsl.ji</td>
<td>PASSED</td>
<td>PASSED</td>
</tr>
<tr>
<td>workspace: 4/16/06 build</td>
<td>2006-04-16 19:33:01</td>
<td>jfrey</td>
<td>PASSED</td>
<td>PASSED</td>
</tr>
<tr>
<td>V6.7-branch</td>
<td>2006-04-17 20:51:32</td>
<td>gthai</td>
<td>PASSED</td>
<td>PASSED</td>
</tr>
<tr>
<td>workspace: Fix NO_.DNS for std_univ</td>
<td>2006-04-17 19:31:00</td>
<td>jfrey</td>
<td>PASSED</td>
<td>PASSED</td>
</tr>
<tr>
<td>V6.7-branch</td>
<td>2006-04-17 20:39:56</td>
<td>bgietzel</td>
<td>PASSED</td>
<td>PASSED</td>
</tr>
<tr>
<td>workspace: All happy now</td>
<td>2006-04-19 20:19:00</td>
<td>jfrey</td>
<td>PASSED</td>
<td>PASSED</td>
</tr>
<tr>
<td>workspace: Globus ext and fix windows</td>
<td>2006-04-19 23:28:27</td>
<td>jfrey</td>
<td>PASSED</td>
<td>PASSED</td>
</tr>
</tbody>
</table>
Users of BaT Facility

› NMI Build & Test Facility was built to serve all NMI projects
› Who else is building and testing?
  • Globus project
  • SRB Project
  • NMI Middleware Distribution
  • Virtual Data Toolkit (VDT)
  • Work in progress
    • TeraGrid
    • NEESgrid
Example I -

The SRB Client
How did it start?

› work done by Wayne Schroeder @ SDSC
› started gently; took a little while for Wayne to warm up to the system
  • ran into a few problems with bad matches before mastering how we use prereqs
  • Our challenge: better docs, better error messages
  • emailed Tolya with questions, Tolya responded “to shed some more general light on the system and help avoid or better debug such problems in the future”
› soon he got pretty comfortable with the system
  • moved on to write his own glue scripts
  • expanded builds to 34 platforms (!)
### Run Results

<table>
<thead>
<tr>
<th>ID</th>
<th>Result</th>
<th>User</th>
<th>Type</th>
<th>Project</th>
<th>Comp</th>
<th>Start</th>
<th>Duration</th>
<th>Description</th>
<th>Platforms</th>
<th>Archive</th>
</tr>
</thead>
<tbody>
<tr>
<td>22929</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-26-2006 17:43</td>
<td>01:22:46</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>33</td>
<td>-</td>
</tr>
<tr>
<td>22930</td>
<td>Removed</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-19-2006 23:05</td>
<td>117:57:28</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>34</td>
<td>-</td>
</tr>
<tr>
<td>22931</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-19-2006 22:45</td>
<td>00:14:35</td>
<td>3.4.0 Client-Only</td>
<td>ppc_ax_5.3</td>
<td>-</td>
</tr>
<tr>
<td>22932</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-19-2006 22:16</td>
<td>00:20:12</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>ppc_ax_5.3</td>
<td>-</td>
</tr>
<tr>
<td>22933</td>
<td>Failed(1)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-19-2006 18:04</td>
<td>In Progress</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>ppc_ax_5.3</td>
<td>-</td>
</tr>
<tr>
<td>22934</td>
<td>Failed(1)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-18-2006 20:17</td>
<td>20:47:01</td>
<td>3.4.0 Client-Only</td>
<td>34</td>
<td>-</td>
</tr>
<tr>
<td>22935</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-18-2006 19:30</td>
<td>21:23:32</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>34</td>
<td>-</td>
</tr>
<tr>
<td>22936</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-18-2006 17:08</td>
<td>00:02:26</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>hppa_hpu_11</td>
<td>-</td>
</tr>
<tr>
<td>22937</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-18-2006 15:03</td>
<td>00:21:13</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>hppa_hpu_11</td>
<td>-</td>
</tr>
<tr>
<td>22938</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-18-2006 15:04</td>
<td>00:02:24</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>alpha_osf_5.1</td>
<td>-</td>
</tr>
<tr>
<td>22939</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 22:12</td>
<td>00:50:49</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>22940</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 22:10</td>
<td>00:04:50</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>alpha_osf_5.1</td>
<td>-</td>
</tr>
<tr>
<td>22941</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 22:08</td>
<td>00:16:06</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>ppc_ydl_3.0</td>
<td>-</td>
</tr>
<tr>
<td>22942</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 22:07</td>
<td>00:33:56</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>hppa_hpu_11</td>
<td>-</td>
</tr>
<tr>
<td>22943</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 22:04</td>
<td>00:14:34</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>alpha_rh_7.2</td>
<td>-</td>
</tr>
<tr>
<td>22944</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 21:44</td>
<td>00:38:56</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>22945</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 21:32</td>
<td>00:06:03</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>22946</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 21:07</td>
<td>00:11:41</td>
<td>test SRB build</td>
<td>sun4u_sol_5.9</td>
<td>-</td>
</tr>
<tr>
<td>22947</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 21:04</td>
<td>00:22:06</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>22948</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 20:53</td>
<td>00:35:52</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>11</td>
<td>-</td>
</tr>
<tr>
<td>22949</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 20:41</td>
<td>In Progress</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>ins_6.5</td>
<td>-</td>
</tr>
<tr>
<td>22950</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 19:39</td>
<td>00:02:06</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>22951</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 19:22</td>
<td>00:05:55</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>ppc_ax_5.2</td>
<td>-</td>
</tr>
<tr>
<td>22952</td>
<td>Failed(2)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 18:13</td>
<td>00:22:42</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>ppc_ax_5.2</td>
<td>-</td>
</tr>
<tr>
<td>22953</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 17:48</td>
<td>00:37:20</td>
<td>Prerelease 3.4.1 Client-Only</td>
<td>x86_rh_9</td>
<td>-</td>
</tr>
<tr>
<td>22954</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-14-2006 17:32</td>
<td>00:12:15</td>
<td>test SRB build</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>22955</td>
<td>Failed(3)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-13-2006 23:15</td>
<td>00:01:49</td>
<td>test SRB build</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>22956</td>
<td>Removed</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-13-2006 23:09</td>
<td>143:19:12</td>
<td>test SRB build</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>22957</td>
<td>Failed(3)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-13-2006 22:58</td>
<td>00:01:26</td>
<td>test SRB build</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>22958</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-13-2006 19:59</td>
<td>00:10:23</td>
<td>test SRB build</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>22959</td>
<td>Failed(4)</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-13-2006 17:39</td>
<td>00:03:31</td>
<td>test SRB build</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>22960</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-11-2006 23:44</td>
<td>00:04:08</td>
<td>test SRB build</td>
<td>x86_slc_3</td>
<td>-</td>
</tr>
<tr>
<td>22961</td>
<td>Complete</td>
<td>schroed</td>
<td>BUILD</td>
<td>srb</td>
<td>srb</td>
<td>Apr-10-2006 19:04</td>
<td>00:03:44</td>
<td>test SRB build</td>
<td>x86_slc_3</td>
<td>-</td>
</tr>
</tbody>
</table>
Where we are today

After ten days (4/10-4/20) Wayne got his builds ported to the NMI BaT facility and after less than 40 runs he reached the point where with “one button” the SRB project can build their client on 34 platforms, with no babysitting. He also found and fixed a problem in the HP-UX version ...
Example II - The VDT
What is the VDT?

› A collection of software
   • Common Grid middleware (Condor, Globus, VOMS, and lots more…)
   • Virtual data software
   • Utilities (CA CRL update)
   • Configuration
   • Computing Infrastructure (Apache, Tomcat, MySQL, and more…)

› An easy installation mechanism
   • Goal: Push a button, everything you need to be a consumer or provider of Grid resources just works
   • Two methods:
     • Pacman: installs and configures it all
     • RPM: installs subset of the software, no configuration

› A support infrastructure
   • Coordinate bug fixing
   • Help desk
   • Understand community needs and wishes
What is the VDT?

- A highly successful collaborative effort
  - VDT Team at UW-Madison
  - VDS (Chimera/Pegasus) team
    - Provides the “V” in VDT
  - Condor Team
  - Globus Alliance
  - NMI Build and Test team
  - EDG/LCG/EGEE
    - Testing, patches, feedback...
    - Supply software: VOMS, CEmon, CRL-Update, and more...
  - Pacman
    - Provides easy installation capability
  - Users
VDT Supported Platforms

- RedHat 7
- RedHat 9
- Debian 3.1 (Sarge)
- RedHat Enterprise Linux 3 AS
- RedHat Enterprise Linux 4 AS
- Fedora Core 3
- Fedora Core 4
- ROCKS Linux 3.3
- Fermi Scientific Linux 3.0
- RedHat Enterprise Linux 3 AS ia64
- SuSE Linux 9 ia64
- RedHat Enterprise Linux 3 AS amd64
VDT Components

› Condor
› Globus
› DRM
› Clarens/jClarens
› PRIMA
› GUMS
› VOMS
› MyProxy

› Apache
› Tomcat
› MySQL
› Lots of utilities
› Lots of configuration scripts

And more!
VDT Evolution

Number of major components

Jan-02 May-02 Sep-02 Jan-03 May-03 Sep-03 Jan-04 May-04 Sep-04 Jan-05 May-05 Sep-05 Jan-06

- VDT 1.0
- Globus 2.0b
- Condor-G 6.3.1
- VDT 1.1.8
  Adopted by LCG
- VDT 1.1.3, 1.1.4 & 1.1.5, pre-SC 2002
- VDT 1.1.11
  Grid2003
- VDT 1.3.0
- VDT 1.3.7 & 1.3.8
  For OSG 0.4
- VDT 1.2.0
- VDT 1.3.0
- VDT 1.3.7 & 1.3.8
  For OSG 0.4

Open Science Grid
VDT’s use of NMI

› VDT does about 30 software builds per VDT release, using NMI build and test facility

› Each software build is done on up to six platforms (and this number is growing)

› Managing these builds would be very difficult without NMI
Build & Test Beyond NMI

We want to integrate with other, related software quality projects, and share build/test resources...

• an international (US/Europe/China) federation of build/test grids...
• Offer our tools as the foundation for other B&T systems
• Leverage others’ work to improve our own B&T service
Exporting the BaT software

Deployments of the NMI BaT Software at international and enterprise collaborators taught us how to make the software portable

- OMII-UK
- OMII-Japan
- EGEE
- Yahoo!
- The Hartford
OMII-UK

• Integrating software from multiple sources
  • Established open-source projects
  • Commissioned services & infrastructure

• Deployment across multiple platforms

• Verify interoperability between platforms & versions

• Automatic Software Testing vital for the Grid
  • Build Testing - Cross platform builds
  • Unit Testing - Local Verification of APIs
  • Deployment Testing - Deploy & run package
  • Distributed Testing - Cross domain operation
  • Regression Testing - Compatibility between versions
  • Stress Testing - Correct operation under real loads

• Distributed Testbed
  • Need a breadth & variety of resources not power
  • Needs to be a managed resource - process
NMI/OMII-UK Collaboration

› Phase I: OMII-UK developed automated builds & tests using the NMI Build & Test Lab at UW-Madison

› Phase II: OMII-UK deployed their own instance of the NMI Build & Test Lab at Southampton University
  • Our lab at UW-Madison is well and good, but some collaborators want/need their own local facilities.

› Phase III (in progress): Move jobs freely between UW and OMII-UK BaT labs as needed.
OMII-Japan

- What They’re Doing
  - “…provide service which can use on-demand autobuild and test systems for Grid middlewares on on-demand virtual cluster. Developers can build and test their software immediately by using our autobuild and test systems”
  - Underlying B&T Infrastructure is NMI Build & Test Software
Moving forward:

ETICS & OMII-EU
ETICS: E-infrastructure for Testing, Integration and Configuration of Software

Alberto Di Meglio
Project Manager
Vision: A dependable, reliable, stable grid infrastructure requires high-quality, thoroughly tested, interoperable software middleware and applications.

Mission: Provide a generic service that other projects can use to efficiently and easily build and test their grid and distributed software. Set up the foundations for a certification process to help increasing the quality and interoperability of such software.
ETICS in a Nutshell

- ETICS stands for e-Infrastructure for Testing, Integration and Configuration of Software
- It’s an SSA
- It has been granted a contribution of 1.4 M€
- It has a duration of two years
- The project has started on January 1st, 2006
The ETICS Partners

Build system, software configuration, service infrastructure, dissemination, EGEE, gLite, project coord.

Software configuration, service infrastructure, dissemination

The Condor batch system, distributed testing tools, service infrastructure, NMI

Web portals and tools, quality process, dissemination, DILIGENT

Test methods and metrics, unit testing tools, EBIT
ETICS Objectives

- **Objective 1 (technical)**
  - Provide a comprehensive build and test service especially designed for grid software
  - Support multi-platform, distributed operations to build software and run complex test cases (functional, regression, performance, stress, benchmarks, interoperability, etc)

- **Objective 2 (coordination, policies)**
  - Establish the foundations for a certification process
  - Contribute to interoperability of grid middleware and applications by promoting consistent build and test procedures and by easying the verification of compliance to standards
  - Promote sound QA principles adapted to grid environment through the participation to conferences, workshops, computing training events (GGF, CSC, ICEAGE)
Service Overview

Web Application

Web Service

Report DB
Project DB
Build/Test Artefacts

NMI Scheduler

ETICS Infrastructure

WNs
NMI Client

Clients

Via browser

Via command Line tools
• Web Application layout (project structure)
• Web Application layout (project configuration)
• Preliminary integration of the client with NMI
Project Timeline

- **January 06**
  - Kick-off

- **April 06**
  - gLite 3.0 is built on ETICS

- **May 06**
  - gLite WMS and VOMS tests run on ETICS
  - More contributions are welcome from other interested parties

- **September 06**
  - gLite 3.1/3.2 is fully built and tested on ETICS
  - QA and Project Management tools are available

- **I Review**
  - All-hands meeting (Madison)

- **All-hands meeting (Budapest)**
Long Term Future and Sustainability

- We envision ETICS to become a permanent service after its initial two-year phase.
- As projects start using and relying on it for managing the software development cycle, the ETICS network should get enough “critical mass” to be supported by research and industrial organization as other “commodity” services.
- In addition, we want to propose ETICS as one of the cornerstones of a more permanent international collaboration to establish a European and world-wide grid infrastructure.
The Problem: What software should I use?

• **Software: There is a lot of it!**
  – Tools, Middleware, Applications, …

• **Quality: Variable!**
  – Large professional teams (e.g. EGEE)
  – Small research groups

• **Interoperability: Not a lot of this!**
  – Standards beginning to emerge from GGF/OASIS/W3C
  – Emerging commitment to provide implementations
  – Need compliance suites and verification activity

• **Need information on quality, portability, interoperability, …**
SA2 - Quality Assurance

• Interoperability through Standards Compliance.
  - Repository components will be tested to establish which standards they comply with.
  - Repository components will be tested to establish which components they interoperate with.

• Documented Quality Assurance
  - Functional operation across different platforms and performance.
Solid Base to Build Upon

- **Open Middleware Infrastructure Institute (OMII) UK**
  - Repository of middleware packages (funded & un-funded)
  - [http://www.omii.ac.uk/](http://www.omii.ac.uk/).

- **Globus Alliance**
  - Open source development portal & software repository

- **ETICS**
  - e-Infrastructure for Testing, Integration and Configuration of Software
  - [http://www.eu-etics.org](http://www.eu-etics.org)

- **NMI Build & Test Framework**
  - Condor based infrastructure for reliable builds & testing
Assembling the Components

- Users
  - OMII/gLite
  - ETICS

- Developers
  - OMII/gLite
  - ETICS

- Portal (Download & Reports)
- Testing Scenarios
- Component Repository
- Build Repository

Testing Infrastructure
- OMII
- UWM
- CERN

EU project: RIO31844-OMII-EUROPE
Building our global Build & Test Grid!