IPv6 readiness of batch systems

Francesco Prelz
INFN, sezione di Milano

(and members of the HEPix IPv6 group)

Summary

- Last talk covering this subject in our community: Barbara Krašovec at the EGI TF 2012, Prague
- Will cover the IPv6 readiness state of the following batch systems (roughly in descending order of readiness or at least IPv6-awareness):
  1. IBM/Platform LSF
  2. (HT)Condor
  3. SGE/OGE/UGE
  4. PBS/Torque
  5. SLURM
- IPv6 capabilities in batch systems may indeed be less urgently needed: services on the CE node act as *natural* protocol translators.
  - We assume that no batch-system network communication escapes from the LAN.
- No direct interaction with the developers/posting of tickets was started for this report, but the most recent source code was inspected when available and existing tickets were looked for.

LSF

- Has been reporting IPv6 support in dual-stack mode at least since 2006 and major version 7 (still calls it 'IPng' in the manuals...).
- From the 9.1.2 (stable/Feb 2013) LSF admin manual:
  «You can use IPv6 addresses if you define the parameter `LSF_ENABLE_SUPPORT_IPV6` *in* `lsf.conf`*; you do not have to map IPv4 addresses to an IPv6 format. For the list of platforms on which LSF supports IPv6 addresses, see the Release Notes for IBM Platform LSF for this version.»*
  - But there's no mention of IPv6 in the current release notes... must be a historical leftover: we should probably be assuming that all platforms are supported.
- It is possible to manually configure the precedence of IPv6 addresses over IPv4 addresses (or vice-versa) via the `LSF_DUALSTACK_PREFER_IPV6` configuration variable.
  - The configuration reference manual documents the following restriction: *«IPv4-only and IPv6-only hosts cannot belong to the same cluster. In a MultiCluster environment, you cannot mix IPv4-only and IPv6-only clusters.»*

  ⇒ Dual-stack and IPv4-only hosts can be mixed.
- Bug-reporting now occurs via the traditional IBM support channels (APARs).
- We don't have anyone in our group who *actually* tested dual-stack LSF yet.

(HT)Condor

- Single-stack pool operation (either IPv4 or IPv6) is supported in the current stable branch 8.0.
    - This was tested. IPv6 has to be explicitely enabled by setting `ENABLE_IPV6=True` in the main `condor_config`.
    - For IPv6 nodes: as long as the host `'hostname'` is resolved as an IPv6 address.
- Mixed mode pool operation (where every network endpoint can be associated to multiple addresses) is being developed and tracked in ticket #3982. The goal is to roll this feature out in the 8.3 development series and make it the default so that it will work automatically at most sites.
- Bug tracking system is integrated in the main project Wiki: `https://htcondor-wiki.cs.wisc.edu`.
    - Searching for 'ipv6' returns about 30 tickets and check-ins over the past few years.

SGE/OGE/UGE

- Plans to add IPv6 support voiced for > 10 years. Full IPv6 support seems to be in the plans for release 8.3, due at end of 2014.
  From the latest UGE 8.1.7 release notes:
  «All service components running on Univa Grid Engine hosts require a IPv4 network that is correctly setup. IPv6 is currently not supported».
    - A quick inspection of the binaries shows that the IPv4 protocol is still hardcoded everywhere, as it is in available open-source versions:
      `sock = socket(AF_INET, SOCK_STREAM, 0);`
- Beyond that, it's hard to understand the plans of UNIVA: did GridEngine really become completely closed-source ("Univa Grid Engine Open Core" on github seems to have stopped in 2012, along with Open Grid Scheduler)?
- Only active fork, with the only public, populated and recently updated issue tracker : Son of Grid Engine.
    - *«This effort precedes Univa taking over gridengine maintenance and subsequently apparently making it entirely proprietary, rather than the originally-promised 'open core'».*
    - Bug tracking system: https://arc.liv.ac.uk/trac/SGE/
        - Transfer of the original Jan. 2012 Gridengine 'Issuezilla' item #2870 on IPV6 support: ticket #620.
          Last modified 4 years ago.

PBS/Torque

- A SVN 'ipv6' branch for Torque was started in 2007, and died.
- We received reports from at least three different groups who attempted to start a dual-stack Torque cluster and failed.
- Very few IPv6/INET6 matches in the source for the latest Torque version, 4.2.8:
    - One address storage tree in `/src/lib/Libutils/u_tree.c` has only room for IPv4 addresses
    - A `IPV6_V6ONLY` preprocessor option, meant to force usage of IPv6, is only referenced in `src/resmom/mom_inter.c` and never used in the configure script.
    - Elsewhere in the code, `getaddrinfo` with `NULL` hints is assumed to return only IPv4 results:
      ```
      if (pbs_getaddrinfo(hostname, NULL, &addr_info) == 0)
      sai = (struct sockaddr_in *)addr_info->ai_addr;
      ```
    - `AF_UNSPEC` is used for a minority of listening sockets.
- ⇒ Porting effort is presumably large, networking code is scattered throughout.
- No mention of IPv6 on the PBS Professional/PBSworks side either.
    - From the fixed-bug list of release 12.0:
      «245150 Installation failure if system has some interface with IPv6 address even though PBS is using only it's IPv4 network»
- Bug tracking system: adaptivecomputing.com support portal (no publicly accessible IPv6 issues)

SLURM

- The status is still the same as reported on the slurm-devel mailing list in May and December 2012: *«adding support should be straightforward. Almost all of the code needing to be changed is in `src/common/slurm_protocol_socket_implementation.c`».*
- IPv4 is still flatly hard-coded everywhere as of the latest version 14.03.3-2:
  ```
  #define AF_SLURM AF_INET
  ```
- At least `src/common/util-net.c` would need to be changed as well:
  ```
  if (family != AF_INET) { errno = EAFNOSUPPORT; return(NULL); }
  ```
- Bug tracking system: `http://bugs.schedmd.com/`
    - No items matching 'ipv6' so far.