C++17 AND LIBSTDC++ ABI

Ceph has switched over to C++17 in mimic. To build Ceph on old distros without GCC-7, it is required to install GCC-7 from additionary repos. On RHEL/CentOS, we are using devtoolset-7 from SCLs for building Ceph. But devltoolset-7 is always using the old ABI even if `_GLIBCXX_USE_CXX11_ABI=1` is defined. So, on RHEL/CentOS, the old implementations of `std::string` and `std::list` are still used. In other words, `std::string` is still copy-on-write, and `std::list::size()` is still O(n) on these distros. But on Ubuntu Xenial, Ceph is built using the new ABI. So, because we are still using libstdc++ and devtoolset for building packages on RHEL/CentOS, please do not rely on the behavior of the new ABI or the old one.

For those who argue that “GCC supports dual ABI!”, here comes the long story. The problem is in the system shared library and `libstdc++_nonshared.a` model. If some symbol is exported from the system shared library, we must use that, and cannot override it. Also, the dual ABI support requires several of the system shared library symbols to behave differently (e.g. for locale facets, need to register twice as many, one set for old ABI, another for new ABI). So, this leaves us with no options but to stick with the old ABI, if we want to enable the built binaries to run on old distros where only the libstdc++ with the old ABI is available.