"Ubiquitous Heads," Image courtesy of http://www.lisacollins.me.uk/

UBIQIS (UBIQuitous InStall)

Install Zero Times, Run Everywhere

Steven Brandt[1,2], Matthew Eastman[3], Matthew Kemp[1,2], Brygg Ullmer[1,2]

[1] Center for Computation and Technology, [2] LSU Computer Science Department, [3] Google Inc.

LSU CENTER FOR COMPUTATION & TECHNOLOGY

Abstract

We propose a system for ubiquitous installation of software resources by mapping resource names uniquely to URLs, fetching them securely on demand (file access), and storing them on a special read-only file system. Compiler wrappers are provided to give the appearance of a seamless universal install. Prototype implementations are available for Java, Python, and C++ on Linux, Mac OS X. Versions of software resources can be accessed directly, or via the ``latest'' version.

UBIQIS Simplifies....

- The user environment
  - No ambiguity about where software should go (i.e. /usr/lib, /usr/local/lib, etc.), since everything is installed under the UBIQIS root directory.
  - Simplified environment variables (one element for CLASSPATH, LD_LIBRARY_PATH, PERL5LIB, etc.)
  - Different versions of the same package have different UBIQIS names, and hence different installation paths
- Installation
  - No need to choose a location for an install
  - No need to explicitly install dependencies
  - No need for a sysadmin

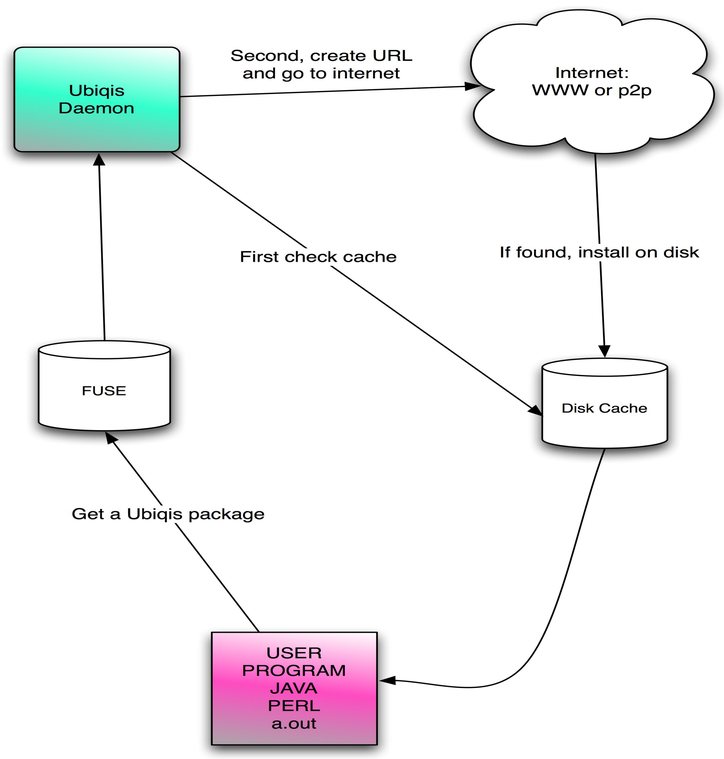
Systems with Characteristic Similar to UBIQIS

–Webstart – One-click install system from web.
- Multiple installations of the same software
- All comes from one source

–Maven – Software build system
- Cache and download packages from web
- Registrar used for location and enforcing uniqueness

–Handle System (http://www.handle.net)

UBIQIS Architecture



UBIQIS Java Example

– User types "cat tst.java"

```
package something.something;
import edu.lsu.cct.ubiq2._tsbrandt.testme.*;
```

...User types: "javac tst.java"

–Compiler looks for a class in the path.... /ubiqis/**edu/lsu/cct/ubiq2/_tsbrandt/testme**/bar/Bar.class

– FUSE intercepts. Constructs the URL: http://cct.lsu.edu/~sbrandt/testme/bar.ubiq

–UBIQIS downloads and installs. Compile succeeds.

–User types: "java something.something.tst"

– Application tries to access Bar.class. It is found in cache.

UBIQIS Naming Convention

–**Example: edu.lsu.cct.ubiq2._tsbrandt.testme => http://cct.lsu.edu/~sbrandt/testme.ubiq**

–The ubiq{num} package element tells us that this is an ubiqis pkg. The {num} tells us remaining significant path elements. This allows complex directory structure inside a package.

–Same UBIQIS package:
- edu.lsu.cct.ubiq2._tsbrandt.testme.Foo.class
- edu.lsu.cct.ubiq2._tsbrandt.testme.bar.Bar.class

–Different UBIQIS packages:
- edu.lsu.cct.ubiq3._tsbrandt.testme.bar.Bar.class
- edu.lsu.cct.ubiq2._tsbrandt.testme.bar.Bar.class

UBIQIS for C++

–Use namespace to describe Ubiqis package.

```
namespace edu { namespace lsu {
  namespace cct { namespace ubiq2 {
    namespace _tsbrandt { namespace { cpphello
  ...
    }}}}}};
// for convenience
namespace cpphello =
  edu::lsu::edu::cct::ubiq2::_tsbrandt::cpphello;
```

–Use the cpp include (or /usr/bin/nm) to trigger FUSE/UBIQIS at compile time, or use the LD_LIBRARY_PATH mechanism to trigger it at run time.

–Stripped down build and simplified build system: creates one of the targets below from all C++ source files in the package.
- lib{pkg}-{build_profile}.so
- lib{pkg}-{build_profile}.a
- {pkg}-{build_profile}.exe
- {pkg}=edu-lsu-cct-ubiq2-_tsbrandt-cpphello

Future Work

- Port a large software package to work in UBIQIS
- Build infrastructure for registrars
- Special Builds (e.g. Atlas)
- Long Builds (progress bars)
- Support Python
- Support for F90
- join us! contact sbrandt@cct.lsu.edu