Provenance for System Troubleshooting

Marc Chiarini

Harvard SEAS

TaPP '11

A Day in the Life...

- Wake up 3am via page from a heartless machine: hot backup has failed.
- Start troubleshooting (in pajamas, thankfully!)
- Log files indicate unable to contact storage appliance, ¾ into backup.
- Storage appliance working fine and reachable now.
- Where to look next? (Coffee first!)

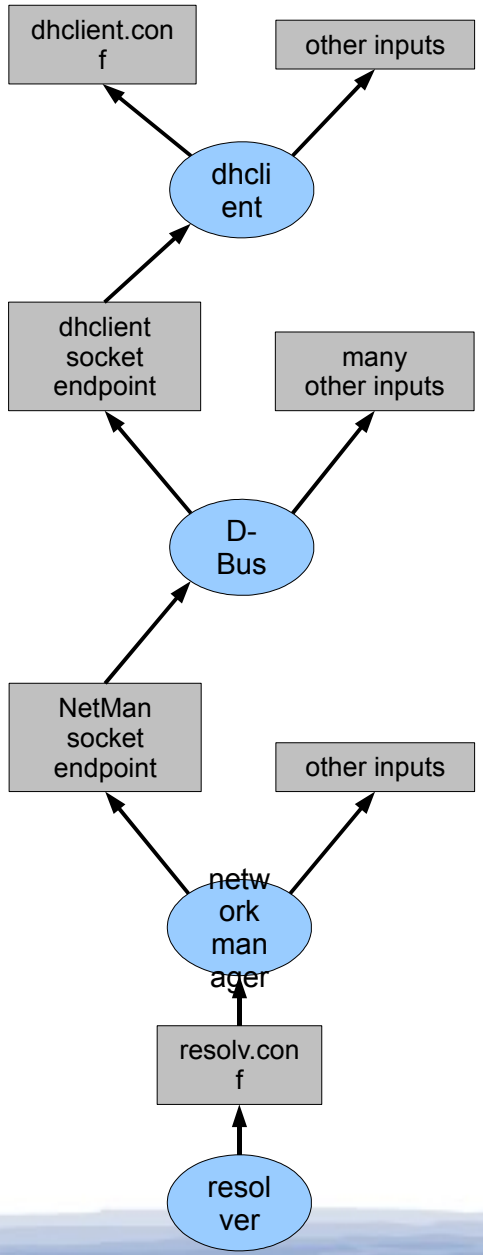
System-level Provenance

- Directed acyclic graph tells us what digital objects interacted during provenance collection, and when.
- Examples:
  - File *F* read by process *P*
  - File *Z* written by *P*
  - *Z* read by process *Q*
  - Pipe *I* written by *Q*
  - *I* read by process *R*

Potential Dependency

- Define *dependency* as the transmission of information from a *passive* object (file, pipe, etc) to an *active* object (process), that is **necessary** to the proper functioning of the process.
- *Transitive dependencies* also exist between active objects.
- For troubleshooting, provenance graph edges represent *potential dependencies*. We don't look at data or programs, so won't talk about

Troubleshooting Example



Graph Reduction

- Real graph is much too large.
- Reduction is necessary to support reasonable queries.
- Want to turn potential dependencies into actual dependencies with high confidence and eliminate non-dependencies in the graph.
- Impossible to identify all true dependencies; would require enumerating all failures.

In Our Favor...

- There are *known* dependencies, e.g., configuration files for system services.
- We can label with low probability, files residing in well-known log directories, e.g., /var/log.
- We can label with high probability, files residing in library directories, e.g. /usr/lib.
- We can label with high probability, files that are opened by a program on every

Other challenges

- Building a tool that improves the sysadmin's mental model of her systems via exploration, documentation, visualization, etc.
- Give the sysadmin an intuitive way to query the provenance graph and limit the scope of query responses (regexps may not cut it!).
- How do we integrate troubleshooting workflow artifacts (e.g., past symptoms and graph query results) with troubleshooting

Questions?

Prototype will be available in late fall 2011.

http://www.eecs.harvard.edu/syrah/pass/

chiarini@seas.harvard.edu