A logging Aspect for Coral

VLM, 25. September 2005

Overview

- 1. What in the world is Coral?
- 2. The Problem
  - Broad
  - Fine
- 3. Goal and Approach
- 4. Related works and links

Coral Overview

- P2P Content Distribution Network
- Uses Distr. sloppy HT to locate stuff
- 160bit opaque identifiers (nodes & content)
- Stores data at nodeID closest to contentID
- Kademlia routing
- Event driven Implementation (libasync - SFS)

The Problem - broad

- 3 running processes:
  - Coral
  - Web Server
  - DNS Server
- Lots of Info, different Types to log
- Distributed Nature of Coral
- Big Code base, difficult to manage logging

The Problem - fine

- Uses complicated logging code spread over all code files for logging
  - Hard to understand
- Different Logger Classes for each of the 3 Processes
- Logged Strings + ints are given to another Logging Process (yet more code) that stores them
- Coral Crawler requests those from all known nodes and generates html

Goal / Approach

- Provide a unified, centralized and easy to understand logging base for coral
- Use AspectC
- No extra process
- Store locally, maybe in DB
- Since Coral is a **CDN**, there might be a way to transfer Stats using Coral itself (instead of crawler)

Related Work / links

- aosd.06: http://aosd.net
- AspectC++: http://www.aspectc.org
- org.apache.jakarta.cactus: http://jakarta.apache.org/cactus
- Andreas Gal, Olaf Spinczyk, Wolfgang Schroeder-Preikschat, "On Aspect-Orientation in Distributed Real-time Dependable Systems", WORDS 2002.

  http://www4.informatik.uni-erlangen.de/~wosch/Publications/2002/words-2002.pdf
- Daniel Mahrenholz, Olaf Spinczyk, Wolfgang Schröder-Preikschat, "Program Instrumentation for Debugging and Monitoring with AspectC++", Symp. on OO RT Dist. Comp. 2002

  http://www4.informatik.uni-erlangen.de/~wosch/Publications/2002/isorc2002.pdf