Dynamic Tracing of NFSv4 with DTrace

Sam Falkner

Sun Microsystems, Inc.

Agenda

- DTrace crash course
- Mapping NFSv4 protocol to DTrace probes
- Examples

What is DTrace?

- Dynamic Tracing, used for debugging, profiling, ...
- No need to recompile, restart, or reboot
- Safe
- System-wide
  - E.g. Follow from applications to libraries to kernel and back

The D Language

- AWK-like syntax
  - A series of conditions and actions
- No looping
- Well-defined (i.e. limited) conditionals
- Cannot modify kernel memory

D Syntax

```
provider:module:function:name
/optional condition/
{
        /* comment: optional action */
        printf("user %d", args[1]->cr_uid);
        globalvar = 1;
        this->localvar = 1;
        self->threadlocal = 1;
        associative["key"] = 7;
        @aggregate["key"] = avg(size);
}
```

DTrace Examples

- When nfs4_read() is called, what's the uid?
  - fbt::nfs4_read:entry{trace(args[3]->cr_uid);}
- How many times is printf() called?
  - pid$target:libc:printf:entry{@pid = count();}

NFSv4 DTrace provider

- Based entirely on the NFSv4 protocol
  - Very few implementation details are exposed
- A probe for every interesting element of the protocol

Syntax

- provider:module:function:name{action}
  - Provider: nfs4c (client) or nfs4s (server)
  - Module: just ignore this
  - Function: covered on next slide
  - Name: start or done

The Function Specifier

- op-compound
- op-<something>
- cb-compound
- cb-<something>
- attr-<something>

Compounds Drive Many Probes

- nfs4c::op-compound:done
  - nfs4c::op-putfh:done
  - nfs4c::op-lookup:done
  - nfs4c::op-getfh:done
    - nfs4c::attr-filehandle:done
      - nfs4c::attr-filehandle-hex:done

Arguments to the Probes

- args[0] is the same on all probes
  - Generic information that all over-the-wire ops have
    - tag, xid, credential, etc.
- args[1] is specific to each probe
  - For functions, the protocol args/res
    - CLOSE4args, GETFH4res
  - For attributes, it's the attribute itself
    - Pointer to a filehandle; a string that's a hex dump of a stateid

Examples

- These examples are untested
- They use interfaces that may change
- They are just examples! :-)

Most Popular tags from client

```
nfs4c::op-compound:start
{
        this->tag = stringof(args[0]->tag);
        @counts[this->tag] = count();
}
```

Server Response Time by Tag

```
nfs4s::op-compound:start
{
        start[args[0]->xid] = timestamp;
}

nfs4s::op-compound:done
{
        this->tag = stringof(args[0]->tag);
        @time[this->tag] = avg(timestamp -
                start[args[0]->xid]);
}
```

Serious Client Debugging

- A problem is reproducible via a test suite
  - But only after running for quite a while...
- Client is misbehaving after it receives a certain response from the server
- The problem-causing response always involves a file named “cthon”
  - But the request is made with the filehandle, and the filehandle isn't predictable

Serious Client Debugging...

```
/*
 * Note transactions from files named
 * “cthon”
 */
nfs4c::attr-component4:start
/args[1] == “cthon”/
{
      thisfh[args[0]->xid] = 1;
}
```

Serious Client Debugging...

```
/*
 * Grab filehandles from response
 */
nfs4c::attr-filehandle4-hex:done
/thisfh[args[0]->xid]/
{
        watchfh[args[1]] = 1;
}
```

Serious Client Debugging...

```
/*
 * When client sends suspicious fh...
 */
nfs4c::attr-filehandle-hex:start
/watchfh[args[1]]/
{
        tracexid[args[0]->xid] = 1;
}
```

Serious Client Debugging...

```
/*
 * start tracing on the response!
 */
nfs4c::op-compound:done
/traceid[args[0]->xid]/
{
        self->traceme = 1;
}

fbt:nfs::entry/self->traceme/{}
fbt:nfs::return/self->traceme/{trace(arg1);}
```