Connectathon ‘97
NFS Version 4:
specific proposals

Mike Eisler
mre@Eng.Sun.Com

OUTLINE

- **In-band security negotiation**
- **New LOOKUP semantics for WebNFS and PCs**
- **Volatile file handles**
- **NFS V4 over TCP**

IN-BAND SECURITY NEGOTIATION

- **Add a new error code, NFS4ERR_SEC_MISMATCH**
  - When an NFS client uses the wrong security, the NFS4ERR_SEC_MISMATCH status will be returned.
  - Add a new procedure, SECINFO to let an NFS client determine the security requirements in band:

```
struct secinfo {
       u_int  flavor;
       opaque flavor_specific<>;
};
struct SECINFO4resok {
       struct secinfo      secinfo_list<>;
}
/* RPCSEC_GSS flavor specific stuff */
struct NFS4_rpcsec_gss_specific {
       opaque gss_oid<>;
       u_int qop;
       enum rpc_gss_service_t service;
}
```

NEW LOOKUP SEMANTICS FOR WEBNFS AND PCS

- WebNFS client must interpret relative URLs and re-compose with base URL to issue another base URL
- The lack of export crossing, case insensitivity causes problems for PCs

```
struct diropargs4   {
       union dirop_fh switch (dirop_base based_on) {
       case RELATIVE:      nfs_fh4 dir;
       default:            void;
       };
       filename4    name;
       u_int        flag;
};
const DIROP4_MCL_CANONICAL = 0x1;
const DIROP4_MCL_NATIVE = 0x2;
const DIROP4_CROSS = 0x4;
const DIROP4_CASE_INSENSITIVE = 0x8;
const DIROP4_CASE_NOTPRESERVED = 0x10;
```

VOLATILE FILE HANDLES

- **Not all file systems, operating systems can generate persistent file handles**
- **Add a “volatile” attribute to file handle that tells clients that persistence assumption is invalid.**
- **Useful for proxy servers, file set migration**

```
struct nfs_fh4 {
       u_int flag;
       opaque data<NFS4_FHSIZE>;
};

const FH4_VOLATILE  = 0x1;
```

NFS V4 OVER TCP

- **How NFS uses (optimizes) over TCP is not specified in NFS Version 2 and Version 3.**
- **As a result:**
  - NFS Servers will still drop requests on floor
    - **Unnecessary for V3, since we have NFS3ERR_JUKEBOX**
  - NFS clients will do retries over good connections
  - RPC level timeouts are frequently too low
  - Problem of recreating connections to use same source address (host/port)

- **Proposal - Define NFS V4/TCP rules as:**
  - NFS V4 Servers MUST NEVER drop requests/responses on floor, unless they
    - **return an error, such as NFS4ERR_JUKEBOX**
    - **break TCP connection (or connection is broken for them)**
  - NFS V4 Clients MUST NEVER retry a request, unless they
    - **receive an error from server for the operation**
    - **receive a disconnect before response arrives**
  - Use NULL procedure to probe connection
  - Employ Eric Werme’s XID generation ideas (Cthon ‘96) so that (well chosen) XIDs with IP addresses are sufficient for uniquely identifying requests
    - **avoids need to re-bind to old port #.**