

2002/12/9
第4回

戦略ソフトウェア-2002-12-9
広域分散ネットワーク上での
ポリシーに基づいたファイル共有を
実現するソフトウェア

江崎研究室
長橋 賢吾 (kenken@wide.ad.jp)

INDEX

- 仕様書について
 - 動作概要
 - Remarque nodeについて
 - 定義すべきメッセージ

Remarque Nodeの仕様(1)

- Remarque Nodeの定義:
 - どのClient Hostであっても、Remarque Nodeになることが可能
 - ただし、Remarque Nodeになるためには、ひとつ以上のpeering(定義は後述)が必要
 - Remarque NodeかClientかどうかのstatusをもつ。Peeringが切断されればClient Hostになる。

Remarque Nodeの仕様(2)

- Remarque Nodeの機能:
 1. Remarque Node同士でのpeering
 2. Client Hostとのinterface
 1. ユーザー(client host)管理
 2. Contents管理
 3. Query処理

Remarque Nodeの仕様(3)

- Peeringの仕様:
 - Remarque Node同士でTCP sessionを確立
 - Port番号は、7600番固定
 - FSM(OPEN->OPENSENT->ACTIVE->ESTABLISH)を持つ
 - ESTABLISHでpeerが確立され、statusがclient->remaqueになる
 - Peerの確立後
 - Remarque nodeが持つファイルポインタ情報を交換する
 - お互い30秒に一度、keep alive messageを出し、生存確認をおこなう
 - Keep aliveが帰ってこない場合は、FSMのOPEN stateに戻る
 - ファイルポインタ情報が更新(追加/更新・削除)された場合、追加/更新はUPDATE、削除はWITHDRAWメッセージをお互いに通知する

Remarque Nodeの仕様(4)

- Client Hostとのinterface
 1. ユーザー(client host)管理
 1. Create User
 2. Logon
 3. logoff
 2. Contents管理
 1. Clientの処理
 2. Remarque nodeでの処理
 3. Query処理

ユーザー管理(1)

- Create User (initial connection from client)
 - Client:
 - FQDNでRemarque nodeを指定
 - ユーザ作成要請メッセージを送出する
 - Remarque nodeからuser-id/passwordを受け取ったら、Remarque node FQDNとuser-id/passwordの組を保存
 - Remarque node:
 - Clientからのユーザ作成要請メッセージを受信
 - Manualでreject/acceptを決定
 - 課金情報等に基づいた自動的な判別もサポートすべき
 - user-id (random文字列+ FQDN)とpasswordを発行し、client-listに追加
 - Clientにuser-id/passwordを通知 (公開鍵/証明書?)

ユーザー管理(2)

- Logon
 - Client
 - Applicationを起動するたびに、1つ以上のRemarque nodeにlogon
 - Logonのために、logon message (w/ user-id/password)をRemarque nodeに通知
 - Remarque node
 - Clientからのlogon messageを受信
 - 自身のClient listと照会
 - 正しい場合は、clientにファイルリスト messageを送信する
 - 間違っていた場合、deny messageを送信

ユーザー管理(3)

- Logoff
 - Remarque nodeでもclientからでもlogoffできる
 - Remarque nodeからのlogoff
 - Clientに対してもkeep alive(30秒に一度)
 - Keep aliveがn回失敗した場合は、clientがleaveしたとみなし、ファイルリストから該当するuser-idに関するリストを削除する
 - Clientからのlogoff
 - Logoff messageをRemarque nodeに送信
 - Remarque nodeはlogoff messageを受信し、ファイルリストを削除して、logoff ok messageをclientに送信

Contents管理(1)

- コンテンツ情報登録の流れ(1):
 - Logonすると、clientは共有フォルダ中のファイルリストをRemarque nodeに登録
 - 登録するために、contents update messageを送信
 - 登録する情報
 - User-id
 - ファイル名
 - ファイルサイズ
 - (ファイルのハッシュ値、ファイル名が異なっても同じファイルを発見できる?)

Contents管理(2)

- コンテンツ情報登録の流れ(2):
 - ファイルリストを受信したRemarque Nodeは、自身の保持しているファイルリストに登録する
 - Clientは、自身のファイルリストを監視し、updateされたら、update messageをRemarque Nodeに送信する
 - 削除の場合は、withdraw messageを同様にRemarque nodeに送信する

Query処理

- Query処理の流れ:
 - Clientが取得したいファイル名を入力して、検索
 - 検索語を、上位のRemarque Node(≥ 1)に通知する
 - Remarque Nodeは、自身のfile listを検索する
 - 部分的にhitするfile情報をclientに通知する

定義するメッセージ(1)

- Peering

| メッセージ名 | パラメータ | 機能 |
|----------------|-------------------------|--------------|
| PEER_OPEN | なし | connection確立 |
| PEER_UPDATE | file情報(最大100 lists/msg) | file情報の追加・更新 |
| PEER_WITHDRAW | file情報(最大100 lists/msg) | file情報の削除 |
| PEER_KEEPALIVE | なし | (30秒に一度)生存確認 |

定義するメッセージ(2)

- ユーザー管理

| メッセージ名 | パラメータ | 方向 | 機能 |
|-----------------|---|-----|----------------------|
| USR_REQ_ADD_NEW | FQDN/任意フィールド(名前など) | C→R | ユーザー作成リクエスト |
| USR_RSP_ADD_NEW | APPROVED(w/ user-id,passwd)/REJECT(w/ reason) | R→C | ユーザー作成リクエストへの応答メッセージ |
| USR_REQ_LOGON | user-id/passwd | C→R | LOGON リクエスト |
| USR_RSP_LOGON | APPROVED/REJECT(w/ reason) | R→C | LOGONリクエストへの応答メッセージ |
| USR_KEEPALIVE | なし | R→C | clientへの生存確認 |
| USR_REQ_LOGOFF | usr-id.passwd | C→R | LOGOFFリクエスト |
| USR_RSP_LOGOFF | APPROVED/REJECT(w/ reason) | R→C | LOGOFFリクエストへの応答メッセージ |

定義するメッセージ(3)

- コンテンツ管理

| メッセージ名 | パラメータ | 方向 | 機能 |
|-------------------|--------------------------|-----|-------------------------|
| CNTS_REQ_UPDATE_1 | なし | R→C | LOGON後のファイルリスト要求 |
| CNTS_REQ_UPDATE_2 | file情報(最大100 lists/msg) | C→R | Clientのファイルリスト登録 |
| CNTS_ACK_UPDATE_2 | SUCEED/FAILED(w/ reason) | R→C | CNTS_REQ_UPDATE_2へのack. |
| CNTS_REQ_WITHDRAW | file情報(最大100 lists/msg) | C→R | ファイル削除要求 |
| CNTS_ACK_WITHDRAW | SUCEED/FAILED(w/ reason) | R→C | 削除要求へのack. |

定義するメッセージ(4)

- Query処理

| メッセージ名 | パラメータ | 方向 | 機能 |
|------------|----------------------------|-----|----------|
| QRY_SEARCH | keyword | C→R | サーチリクエスト |
| QRY_RESULT | result (up to 100hits/msg) | R→C | サーチ結果 |

具体的なイメージ(最近考えていること)

- たとえばストリーミング/VoD
 - ClientはRemarque Nodeと契約する
 - 契約すると、Remarque Nodeの持っているcontents listを表示
 - Clientは任意のコンテンツにアクセスする
 - この場合、接続peerは契約した会社(TV会社など)の分散したホスト