

سیستم عامل پیشرفته

محمد داوریناه جزئی

ترم دوم ۹۴-۹۳

گروه مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی فولاد

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

• مسائل مهم در روش RPC

- گره RPC پر استفاده‌ترین روش در Client/Server و سیستم توزیعی است ولی مسائلی دارد که باید توجه شود.
- سیستم توزیع‌شده قرار شد `Transparant` باشد ولی در عمل کمتر سیستمی است که صددرصد اینطور باشد.
- سیستمی که قبلاً تک CPU بوده حالا PRC در آن وارد شده آیا می‌تواند به جای شکل سابق مورد استفاده قرار گیرد!
- متغیرهای سراسری مثل `Errorno` اگر توسط دو پروسس روی دو پروسسور مختلف مورد دستیابی قرار گیرد کار فراب می‌شود.
- ارسال پارامتر آرایه به اندازه نامشخص در زبان‌های مثل C (`Weakly Typed`)، `Stub` نمی‌داند چه سائزی بگیرد هر ما کزیممی هم بگیرد می‌تواند ایجاد اشکال کند (کمبود جا با محدودیت)
- ارسال اشاره گر به یک سافتمان داده پیچیده، ارسال پارامترهای جورواجور مثل `Print f`
- اگر قرار شد اینها را ممنوع کنیم `Transparany` نقص شده است.
- مسئله نوع دیگر در دستورات `Sort` و `Grep` در `Unix` ص ۹۷ هر دو پروسس به عنوان `Client` از `Server` می‌فوانند و روی `Server` می‌نویسند.
- در دستور `Pipe` شده از این دو، یکی باید `Client` و یکی `Server` باشد که قاعده بالا نقص می‌شود.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- پس Pipe کردن به فرم ص ۹۷ امکان پذیر نیست
- راه حل Read Driven شکل P۲b-P۲ ص ۹۸ تا مرحله P3 کار می‌کند همه Client هستند ولی برای انتقال نتیجه به فروبی Server نمی‌تواند Client شود.
- همین مسئله در راه حل Write در شکل P۲c-P۲ است برای دادن ورودی به P1
- راه حل‌های ارائه شده مثل Dual Server که همزمان بفوائد و بنویسد دارای Over Head زیاد است
- حالت دیگر وقتی است که ترمینال یک Server است و کاراکتری می‌فرستد به Process ولی وقتی بفوائد CntrlC یا Break بفرستد باید بشوند Client که فلاف است.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

• ۵- ارتباط گروهی

- در مواردی چون حالتی که چند Server یکجا به منظور تحمل فضا موازی کار می‌کنند یک پیغام مشتری به همه باید ارسال شود.
- RPC این امکان را ندارد مگرى چند RPC مستقل، روشهای دیگری اینجا بحث می‌شود.
- یک ارسال کننده و چند گیرنده (One to Many) به جای یک به یک شکل. ۳-۲ ص ۹۹
- گروهها دینامیک هستند، Processها می‌ایند و می‌روند و گروهها ایجاد و منحل می‌شوند.
- یک Process می‌تواند همزمان عضو چند گروه باشد.
- پس مکانیزم دقیقی برای مدیریت آنها نیاز است.
- در کامپیوتر مثالهای دیگر گروهها مثل Newsgroup هم هست که قوانین ساده‌تری دارند.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- گروه یک موچودیت مستقل است، از یک Call تا Call بعدی تغییر می‌کند و ارسال کننده از اینها بی‌خبر است.
- یگونگی پیاده‌سازی به سفت‌افزار بستگی دارد.
- یک راه حل Set کردن یک بیت فاص High or der که گروهی از ماشینها به آن گوش کنند.
- به این روش Multicasting گویند، ساده است و هر گروه یک آدرس فاص دارد.
- روش دیگر Broadcast است که پیغام‌های فاص (با شماره صفر) به همه ماشینها میرود.
- کارائی کمتر، ماشینها برای هر پیام یک وقفه می‌گیرند پیغام را چک می‌کنند اگر مربوط به آنها نباشد توجهی نمی‌کنند، مقدارای زمان تلف می‌شود.
- در نهایت (دو روش بالا نباشد) می‌توان پیام را n بار برای n نفر ارسال کرد، کارائی کمتر ولی امکان‌پذیر.
- به این حالت Unicastig گویند.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- مفاهیم طراحی
- وجود مفاهیم عمومی **Blocking** و **Buffering** و مفاهیم خاص **Grouping**.
- گروه‌های بسته (عدم امکان ارسال از بیرون) و گروه‌های باز (امکان ارسال از بیرون) شکل ۳۱-۲ ص ۱.۱
- تصمیم‌گیری در زمان شکل‌گیری گروه، پروسس‌های موازی در یک بازی شطرنج گروه بسته است.
- **Fileserver** گروه باز هم ارتباط داخلی و هم ارتباط از بیرون.
- گروه سلسله‌مراتبی (و چون یک یا چند طبقه هماهنگ‌کننده) گروه **Peer** (هم‌بطور مستقل و با هم) شکل ۳۲-۲ ص ۱.۲
- در سلسله‌مراتبی هماهنگ‌کننده مسئول تقسیم کار بین سایر اعضا، از کار بیفتد کل سیستم می‌فوتد ولی تصمیم‌گیری ساده است.
- در **peer** تصمیم‌گیری‌ها نیاز به رأی‌گیری دارد، ولی از کار افتادن یک سیستم کار را فراب نمی‌کند.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- عضویت: نیاز به Group Server برای مدیریت گروه‌ها و عضویت‌ها و سایر موارد، نگهداری یک Database از گروه‌ها.
- فطر همیشگی در شکست این Server، نیاز به سافت‌ن مجدد اطلاعات و از دست دادن کارهای در حال انجام.
- روش دیگر مدیریت توزیع شده، عضویت‌ها و فروج‌ها به همه اعضا اطلاع داده شود.
- نکته مهم همگام بودن Join و Leave با ارسال و دریافت پیام است.
- به محض Join می‌تواند پیام بفرستد و باید همه پیام‌های گروه را بگیرد و برعکس آن به محض Leave.
- یک راه تبدیل ایندو عمل به ارسال پیام به اعضا است.
- نکته دیگر توقف یک عضو گروه (Leave بدون اختیار) است که بقیه بطور تجربی متوجه می‌شوند.
- نهایتاً اگر تعداد زیادی عضو توقف کردند گروه بی‌معنی می‌شود.
- راهی برای دوباره‌سازی گروه نیاز است. چه کسی شروع کند، اگر دوتا یا بیشتر همزمان شروع کردند چه؟ پروتکل باید قابلیت حل آن را داشته باشد.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- آدرس‌دهی گروه
- هر گروه می‌تواند یک آدرس داشته باشد مثل هر Process.
- در Multicast پیام به اعضا و نه کس دیگر ارسال می‌شود.
- در Broadcast پیام به کلیه ماشینها می‌رود، Kernel تشخیص می‌دهد مال چه پروسی است و برای آن می‌فرستند. (User فقط یک پیام می‌فرستند بقیه با os است.)
- اگر مال هیچ پروسی نیست Discard می‌شود. (User فقط یک پیام می‌فرستند بقیه با os است.)
- در Unicast، Kernel فرستنده عمل ارسال را برای همه اعضا تکرار کند. شکل ۳۳-۲ ص ۱۴.
- راه دیگر اینکه ارسال کننده لیستی از آدرس IP گیرنده‌ها را بدهد (اشاره گر به لیست بعنوان پارامتر)
- مشکل عدم Transparency هر User باید بداند اعضا هر گروه چه کسانی هستند و آن لیست را بهنگام هم بکند (هنگام Join و Leave)
- سومین روش Predicate Addressing است که یک عبارت بولی همراه با پیام می‌رود و به همه.
- در صورتیکه جواب T بود ماشین گیرنده قبول پیام می‌کند و گرنه رد پیام (مثلا به کلیه ماشینها با 4M حافظه آزاد)

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- فرمانهای اولیه ارسال و دریافت
- در RPC ارسال و دریافت (Req/Reply) دو طرفه و یک به یک است، چند به یک به چند امکان‌پذیر نیست (گروهی)
- راه عمومی بر گشتن به ارتباط یک طرفه است، امضار صریح برای ارسال و دریافت.
- اگر برای یک به یک RPC داریم برای گروه از روش دیگر استفاده می‌شود.
- اگر باری یک به یک هم Receive و Send داریم همین برای گروه هم استفاده می‌شود.
- در حالت دوم، برای Send دو پارامتر داریم یکی لیست گیرنده‌ها یا آدرس گروه و دومی خود پیام.
- مسائل Buffering، Reliability و blocking هم کماکان وجود دارند و در زمان طراحی تصمیم‌گیری می‌شوند.
- Receive منتظر پیام است و ممکن است بلوک هم باشد.
- در حالت مفلوط می‌تواند پیام یک به یک بگیرد و یا چند ولی بعضی سیستم‌ها اینرا با دستورات متفاوت انجام می‌دهند.
- دریافت مستقل از ارسال است پس ارسال‌کننده باید با `getreply` تکراری پاسفا یکی یکی را جمع‌آوری کند.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

Altomiuty یا Atomil Broadcant

- پیغام ارسالی یا به همه برسد یا به هیچ کس نرسد، برنامه‌نویسی توزیعی را راحت‌تر می‌کند.
- اگر این نباشد، یک رکورد بعضی جاها ایجاد می‌شود و بعضی جاها نه‌الا عیثمت همه امکان ندارد.
- پیاده‌سازی این روش ساده نیست، نیاز به ack برای هر پیغام داریم.
- اگر همه ماشینها زنده و در حال کار باشند با ack مشکلی نیست ولی آیا همیشه اینطور است.
- سیستم توزیع شده باید تحمل‌پذیر فضا باشد.
- موقعیتی که پیغام‌ها به بعضی‌ها نرسد over run، آنها هم که فبر ندارند، بعدش هم ارسال‌کننده توقف کند
- یک راه ارسال توسط Sender، هر کامپیوتر به محض اولین دریافت ارسال برای بقیه اعضا، گردد بالاخره یک فرمان همه اعضا، زنده پیام را گرفته‌اند. (روشهای بهتر بعد بحث می‌شود).

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- ترتیب پیغام‌ها
- در کنار Atomicity ترتیب پیامها هم اهمیت دارد.
- شکل ۳۱۴-۲ ص ۱۰۸ پروسیسهای . و او ۳ و ۴ یک گروه‌ند.
- قرار است همزمان صفر یک پیغام و ۴ هم یک پیغام بفرستند.
- ترتیبی که ۱ پیغام‌ها را می‌گیرد با ترتیبی که ۳ آنها را می‌گیرد تفاوت دارد.
- مال اگر اینکار update باشد کار فراب می‌شود و شبیه حالتی که پیغام اصلاً نرسد.
- ترتیب بستگی به حالت شبکه دارد و نیاز به یک Semantic قوی برای اینکار داریم.
- راه‌حل ارسال یک پیام به محض شروع به همه و سپس پرداختن به پیام بعدی است.
- به این Global Time Ordering گویند ولی پیاده‌سازی آن همیشه ساده نیست.
- حالت رامت‌تر Consistent Time Ordering است که وقتی دو پیام نزدیک مهم داده شود یکی انتفاع شده کامل می‌شود و بعد دیگری، ممکن است ترتیب کلی برعکس باشد ولی کسی نمی‌داند.
- فقط.....می‌کند که پیغام یکجا به همه می‌رسد.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- گروه‌ها با اعضا مشترک
- با توجه به شکل ۳۵-۲ ص ۱۰۹ ترتیب وصول پیغام‌های از A, D به B, C فرق می‌کند.
- چون از دو گروه مجزا می‌آید GTO اثری روی آن ندارد.
- بعضی سیستم‌ها این موضوع را بین گروه‌های هم پوشان در نظر می‌گیرند و لی کار مشکلی است باید دید در مقابل هزینه آن ارزش دارد و یا نه؟

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- توسعه پذیری
- تعداد زیاد اعضا، تعداد زیاد گروه‌ها، تعداد شبکه‌ها، فواصل زیاد جغرافیائی مواردی هستند که مشکل‌زا می‌باشند.
- شکل ۳۶-۲ ص. ۱۱، ارسال پیام از LAN2، دوبار می‌آید به LAN3 و بعد تکراری می‌چرند به LAN2 دیگر.
- الگوریتم پیچیده‌ای برای این شرایط نیاز است.
- قانون وجود یک پکت روی هر LAN در یک لحظه نقض می‌شود.
- توسعه بعضی الگوریتم‌ها به دلیل طول محاسبات یا عناصر متمرکز برای تعداد زیاد امکان ندارد.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- ارتباط گروهی در ISIS دانشگاه Carnell مثال عملی پر استفاده
- مجموعه نرم‌افزار که روی سر یک سیستم عامل مثلاً UNIX قرار می‌گیرد.
- مبنای این سیستم همگامی (Synchronising) است.
- یک پیغام در زمانی بطول صفر به همه اعضا گروه می‌رسد و همه عمل‌ها به ترتیب انجام می‌شود. شکل
۱۳۷-۲ ص ۱۱۱
- در لحظه M1 پیغام A به B, C, D می‌رسد و در M2 پیغام D به بقیه می‌رسد.
- اینکار در عمل امکان‌پذیر نیست باید تفهیم بدهیم.
- در Loosly Synchronaus ارسال در یک فاصله زمانی انجام می‌شود ولی ترتیب هنوز وجود دارد.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- همه پروتکل‌ها همه پیام‌ها را با یک ترتیب می‌گیرند. شکل ۳۷-۲ ص ۱۱۱
- اینکار عملی است و در بعضی سیستم‌ها باز هم می‌شود تفضیف داد تا کارایی بهتر شود.
- Virtually Synchronons شکل ۳۷-۲ ص ۱۱۱ که در آن کار ساده‌تر است.
- دو پیغام Concarreut از هم مستغند، دو پیغام Causally related یکی از دیگری مشتق می‌شود.
- در اولی ارسال پیغام‌ها ترتیبی نمی‌فواهد ولی در دومی باید به ترتیب باشد.
- برپایه این دو خصوصیت می‌توان ارسال را به ترتیب یا بدون ترتیب انجام داد.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- دستورات اولیه ISIS (پروتکل‌ها)
- ABZAST: ارتباط همگام ضعیف (Loosly) برای Data.
- روش اولیه، استفاده از 2 Phase Commit، ارسال پیام با یک Time Stamp و ارسال جواب هم همینطور.
- پس از وصول همه جواب‌ها، بزرگ‌ترین TS را با پیام Commit به همه ارسال می‌کند.
- دستورات Commit با ترتیب TS به برنامه‌ها ارسال می‌گردد. (گارانتی ارسال پیام به همه با یک ترتیب).
- ترتیب رعایت می‌شود ولی روش پیچیده است و گران.
- CBCAST: ارتباط همگام مجازی (Virtual) برای Data.
- بدلیل پیچیده بودن قبلی بوجود آمد، فقط ترتیب در پیام‌های وابسته به هم را تضمین می‌کند.
- هر پروسس آرایه‌ای از شماره ترتیب به تعداد اعضا، گروه دارد.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- ارتباط گروهی در ISIS دانشگاه Carnell مثال عملی پر استفاده
 - مجموعه نرم‌افزار که روی سر یک سیستم عامل مثلاً UNIX قرار می‌گیرد.
 - مبنای این سیستم همگامی (Synchrang) است.
 - یک پیغام و (زمانی بطول صفر به همه اعضا گروه می‌رسد و همه عمل‌ها به ترتیب انجام می‌شود. شکل
- ۱۳۷-۲ ص ۱۱۱
- در لحظه M1 پیغام A به B, C, D می‌رسد و در M2 پیغام D به بقیه می‌رسد.
 - اینکار در عمل امکان‌پذیر نیست باید تفهیم بدهیم.
 - در Loosly Sgnchronaus ارسال در یک فاصله زمانی انجام می‌شود ولی ترتیب هنوز وجود دارد.