

سیستم عامل پیشرفته

محمد داوریناه جزئی

ترم دوم ۹۴-۹۳

گروه مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی فولاد

روش مدیریت درس

ارزیابی درس

- تکالیف درسی 10%
- مقاله + ارائه در کلاس 15%
- امتحان میان ترم 30% + جایزه
- امتحان پایان ترم 45% + جایزه
- روز حساب دارای نمره اضافی
- اسلایدهای درس روی سایت davarpanahjazi.iut.ac.ir
- مرجع اصلی درس:

Distributed Operating Systems

A S Tanenbaum

- سایر مراجع درس: دریای بیکران اینترنت

فهرست مطالب

- فصل اول مفاهیم اولیه سیستمهای توزیع شده
- فصل دوم ارتباطات در سیستمهای توزیع شده
- فصل سوم همگام‌سازی در سیستمهای توزیع شده
- فصل چهارم سیستمهای توزیع شده. پردازشها و پردازشگرها
- فصل پنجم سیستمهای فایل توزیع شده
- فصل ششم حافظه‌های توزیع شده مشترک

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده

۱- تعریف: یک سیستم توزیع شده در مقابل یک سیستم متمرکز (سنتی) عبارتست از تعدادی کامپیوتر مستقل که از دید استفاده کننده یک کامپیوتر فرض می‌شود.

مثال: کامپیوترهای موجود در یک شرکت، ربات‌های دافل یک کارخانه + کامپیوتر اصلی، کامپیوترهای یک شبکه بانکی

۲- مزایای عمده:

- دلیل اقتصادی، اتصال تعدادی کامپیوتر کوچک ارزانتر و قویتر از یک کامپیوتر بزرگ
- سرعت بالاتر، سرجمع قدرت پردازشی بیشتر از کامپیوتر بزرگ
- ذات توزیعی بودن کارها (شبکه فروشگاه یا بانک)، کارها یا بازی‌های گروهی توزیعی
- قابلیت اعتماد، فرآبی یکی از ده ماشین ۱/۱۱ از بازدهی کم می‌کند.
- قابلیت رشد تدریجی
- فاصله در ش ۱-۱ ص ۵

مزایای سیستم توزیع شده نسبت به PCهای مستقل:

اشتراک داده‌ها، اشتراک دستگاه‌ها، ارتباط بین انسان‌ها، انعطاف‌پذیری (ش ۱-۲ ص ۷)

معایب عمده: نیاز به نرم افزارهای قوی، امکانات شبکه و توسعه آن، امنیت اطلاعات (ش ۳-۱ ص ۸)

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده (ادامه)

۳- سفت‌افزار:

- SISD (تک پردازنده‌ای)، SIMD (پردازنده‌های برداری مثل بعضی سوپر کامپیوترها)، MISD (نداریم)، MIMD (چند پردازنده‌ای یا شبکه) ش ۱-۴ ص ۹
- Tightly Coupled در مقابل Loosely Coupled، سرعت در حد فاصله (parallel systems) در مقابل سرعت در حد دستگاه انتقال (distributed systems) در مقابل Bus
- چند پردازنده‌ای + BUS ش ۱-۵ ص ۱۱، ارتباط با حافظه - حافظه مرتبط - ۳۲ تا ۶۴ فضا دیتا و آدرس روی یک BUS + مثلاً حافظه کنترل - همه امور موازی - شلوغی - (حافظه Cache - Coherent) و موضوع - hit rate مسئله - incoherency استفاده از Snoopy و Write Through
- چند پردازنده‌ای + سوئیچ ش ۶-۱ ص ۱۲: Crosspoint 12: دسترسی همزمان، مجذور n، گران، (Omega گران، $\log_2(n+1)$) کند، مناسبه پ ۲ ص (۱۳، NUMA حافظه محلی و بیرونی، پیچیدگی بایگزینی نرم‌افزار و داده برای جهت دهی به سمت درونی)
- چند کامپیوتر + BUS ش ۱-۷ ص ۱۴: انتقال فقط بین CPU ها - ترافیک چند توان ۱۰ کمتر - شبکه ضعیف‌تر
- چند کامپیوتر + Switch ش ۱-۸ ص ۱۴: Grid طول مسیر زیاد جذر (n، Hypercube مثال CPU 16384، طول مسیر کمتر، رشد مسیر و تعداد اتصالات $\log n$)

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده (ادامه)

۴ - نرم‌افزار برای چند پردازنده (مهم‌تر از سخت افزار):

- **Loosly Coupled:** چند کامپیوتر مستقل ولی تا حدی مرتبط، شبکه LAN، اشتراک بعضی منابع مثل پایگر و دیسک، قطع ارتباط، امکان ادامه کار و حمل دیسک (هنوز مرتبط ولی...)
- **Tightly Coupled** (یک برنامه شطرنج که توسط تعدادی CPU اجرا می‌شود اجرای هر صفحه توسط یک CPU)
- با پنهان کردن نحوه ارتباط چند پردازنده از برنامه‌نویس، ۴ حالت ترکیب نرم‌افزار و سخت‌افزار (به جای ۸ حالت)
- الف LC - در سخت‌افزار و نرم‌افزار: WS‌های مستقل مرتبط با LAN، Rlogin، Rcopy، دستی و محدود (user همه چیز را می‌داند)
- ب Client/Server - ش ۹-۱ ص ۱۷، ایجاد دایرکتوری مجازی ش ۱-۱ ص ۱۸، نیاز به زبان مشترک برای انتقال پیغام‌ها، مدیریت با سیستم‌عامل

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده (ادامه)

(1) پ - سیستم عامل شبکه (LC در سفت‌افزار و نرم‌افزار)، استقلال در هر دو زمینه فقط انتقال پیغام‌ها، کاربر از همه چیز مطلع است.

(2) ت - نرم‌افزار TC (روی سفت‌افزار LC یعنی کاربر کل سیستم را یک ماشین فرض کند:

سیستم توزیعی واقعی Uniprocessor Virtual یا Single System Image هنوز به طور کامل تحقق نیافته

• فاصله فواصل مهم NOS : مکانیزم یکسان ارتباط بین پردازنده‌ها (IPC)، مدیریت پردازش‌ها به طور یکسان، استفاده از یک مجموعه مشترک System call ، یکسانی سیستم فایل در همه جا، داشتن Kernel های یکسان، سیستم فایل عمومی، دادن اختیارات محلی به Kernel های محلی

• ث - نرم‌افزار و سفت‌افزار هر دو) TC سیستم‌های اشتراک زمانی چندپردازنده‌ای (- ماشین DB

– چند CPU با حافظه مشترک، ش 1-11 ص 20 ، B تمام شود و D بخواهد شروع شود، ورود به بخش بحرانی سیستم‌عامل برای عدم انتخاب پردازش بعدی توسط دو پردازنده

– انتخاب CPU ها مهم نیست فقط با در نظر گرفتن Cache بابه‌ها شدن کمی وقت می‌گیرد مگر CPU که قبلاً روی آن اجرا شده

– انتظار مشغول CPU در صورت کم بودن زمان انتظار در مقابل زمان Switch

– فاصله روشها در شکل 1-12 ص) 22 توضیح

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده (ادامه)

5. مفاهیم طراحی

• شفافیت (به کاربر) Transparency

مثال: کامپایل مجموعه بزرگی از فایل‌ها روی چند ماشین بدون اینکه کاربر بداند و بفهمد دو سطح شفافیت: گول زدن user ساده‌تر از گول زدن برنامه نویس (شکل ۱-۳ ص ۲۳)

— Location: اسم‌ها معین کننده محل‌ها نباشند.

— Migration: جابه‌جایی منبع باعث تغییر نام نشود (انتقال News در شکل ۱-۱ که باعث تغییر نام شده)

— Replication: ایجاد کپی از فایل‌ها بدون اینکه کاربر بداند (سیستم بانک)

— Concurrency: جلوگیری از استفاده همزمان بدون اینکه کاربر بداند (استفاده از قفل)

— Parallelism: بکارگیری موازی مثلاً CPUها در موارد لزوم بدون دفالت برنامه نویس، مشکل دفالت، (نقطه پایان و مدینه فاضله!)

عدم نیاز به شفافیت در بعضی موارد مثل چاپ یک فایل روی یک چاپگر راه دور - رایگان - دارای جایزه، رنگ و بوی فوش!!

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده (ادامه)

• انعطاف‌پذیری Flexibility

احتمال متروک شدن مفاهیم قابل قبول فعلی در آینده نزدیک، وجود دو طرز فکر ش ۱-۱۴ ص ۲۵

— Monolithic Kernel یا هسته حجج : همه سرویس‌های لازم داخل Kernel. سیستم فایل، مدیریت حافظه، مدیریت پروسس‌ها و ...

— Micro Kernel: حداکثر کوچک کردن آن و انجام کارهایی نظیر IPC، مقدار کمی مدیریت حافظه، مدیریت پروسس سطح پایین و I/O سطح پایین و انتقال مواردی چون عملیات سیستم فایل، دایرکتوری، مدیریت کامل پروسس‌ها، ارسال درخواست به یک Server و دریافت جواب پس از انجام کار به سطح کاربر

— استفاده از Micro باعث انعطاف می‌شود. دسترسی یکنوافت به سرویس‌ها برای همه وجود دارد. می‌توان بدون توقف سیستم سرویس‌ها را تغییر داد و سرویس‌های مناسب را اضافه کرد که اینها در دیگری همه ثابت هستند. در اینجا زمان انتظار جهت انجام سرویس‌ها در سطح بالاتر قابل گذشت است). چاپ سنجاقک و فنچ به جای دایناسور روی جلد User(ها می‌توانند روتین سرویس برای فودشان بنویسند.

— وجود سیستم فایل DOS و UNIX با دو روش متفاوت در یک سیستم (مثال Sun)

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده (ادامه)

• قابلیت اعتماد Reliability

- (جمله لسانی لمپورت) نمی‌توانم کار کنم زیرا یک ماشینی که اصلاً ندیده‌ام فراب است.
- در تئوری Or عناصر ولی در عمل AND شان، ۴ سرور با ۹۵% عملیاتی، ۹۹/۹۹۹۴% کل
- در دسترس بودن از طریق تکرار بلوک‌های سخت و نرم حساس و بحرانی، عدم نیاز به عملکرد همزمان آنها
- حفظ سازگاری کپی‌های اطلاعات (کپی بیشتر = در دسترس بودن بالاتر ولی نیاز به حفظ سازگاری بیشتر)
- امنیت: اشکال مهم و اساسی در تشخیص هویت ارسال کننده پیغام، شماره کارت اعتباری دزدیده می‌شود، تفاوت با سیستم‌های ساده
- تحمل فرابی: امکان ترمیم بعد از فرابی بدون با فبر کردن کاربر، فقط کند شدن روند کار در صورت توقف یک Server بقیه بتوانند کار آن را بازسازی و جبران کنند.

• کارکرد Performance

- داشتن موارد بالا با کارکرد ضعیف در برابر یک سیستم متمرکز فوب نیست.

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده (ادامه)

- صرف وقت برای ارتباطات و انتقال پیغام‌ها که در سیستم متمرکز نیست: Throuout در مقابل Throuput، به کارگیری، شلوغی شبکه
- به حداقل رساندن تعداد پیغام که مغایر است با موازی کار کردن پراسسورها، $a*b+c*d$
- انجام محاسبات کوتاه و وابسته به هم Fine grained parallelism در محل
- ارسال محاسبات طولانی با نقل و انتقال کم Course grained parallelism به سایر پردازنده‌ها
- انجام موازی یک کار توسط دو پردازنده برای ترمیم شکست یکی از آنها (شلوغ کردن شبکه!)
- Bench mark نمی‌تواند ثابت باشد بلکه به نوع برنامه بستگی دارد CPU bound در مقابل I/O bound، محاوره‌ای در مقابل batch، بلادرنگ در مقابل معمولی، . . .

• قابلیت توسعه Scalability

- امکان جایگزینی $P..$ میلیون CPU به جای $P..$ تا که فعلاً هست وجود ندارد.
- نصب ترمینال در کلیه منازل فرانسه برای سرویس تلفن، پست، دسترسی به اطلاعات عمومی، بانک، محل و نقل، هتل و
- انجام متمرکز عملیات مثل Mail server امکان‌پذیر نیست به دلیل حجم زیاد I/O

فصل اول: مفاهیم اولیه سیستم‌های توزیع شده (ادامه)

- جداول اطلاعاتی متمرکز امکان ندارد (نگهداری یک فایل از اطلاعات شخصی ۵ میلیون آدم)
- الگوریتم‌های متمرکز قابل اجرا نیست: هر روشی که اطلاعات از همه جا را در یک جا جمع کند محکوم به شکست است.
- گذشتن هر کاری بر عهده عنصر متمرکز!!
- باید الگوریتم‌های غیر متمرکز که خصوصیات زیر را دارد مورد استفاده قرار گیرد.
 - ۱- هیچ ماشینی اطلاعات کامل از کل سیستم ندارد،
 - ۲- ماشین‌ها بر مبنای اطلاعات محلی خود تصمیم می‌گیرند،
 - ۳- توقف یک ماشین الگوریتم را نابود نمی‌کند،
 - ۴- فرض ضمنی در مورد وجود یک ساعت عمومی سیستم وجود ندارد.
- اهمیت مورد آفر، عدم امکان همگام کردن ساعتها به ویژه در شبکه‌های بزرگ ملی و ... (ف ۳)