بررسی مزایای Linux به‌عنوان سیستم‌عامل server شبکه‌ی نرم‌افزارهای تخصصی نفتی - قسمت دوم

پرستو امامی، شرکت نفت خزر

۱-پایداری و قابلیت اطمینان (در قسمت قبل توضیح داده شد.)

۲- ادامه‌ی مبحث امنیت (Security)

در ادامه‌ی مباحث قسمت قبل[1]، می‌توان نتیجه‌گیری کرد که در شبکه‌های لینوکسی، سیستم‌ها به‌معنای واقعی، سیستم‌هایی چند کاربره هستند و مهم‌تر اینکه برای هر فایل تکی یا هر دایرکتوری می‌توان سطوح دسترسی کاربران و گروه‌های کاربری تعریف کرد و هر کاربر به‌صورت پیش‌فرض دارای یک محدوده‌ی امن اطلاعاتی است.

مطلب حائزاهمیت دیگر در ارتباط با کاربران و گروه‌های کاربری این است که هر کاربر در سیستم لینوکس یک دایرکتوری شخصی به‌نام home خواهد داشت که در آن فضا دارای اختیارات و دسترسی کامل به فایل‌های خود می‌باشد و می‌تواند در آن محیط، به ایجاد و حذف فایل یا عملیات دیگر بپردازد و هیچ‌یک از کاربران یا گروه‌های دیگر به‌جز کاربر ریشه(root) نخواهند توانست به اطلاعات او دسترسی یابند. (نکته‌ی مهم امنیتی)

در سیستم‌های لینوکسی هر کاربری که مالک یا ایجادکننده‌ی یک فایل یا دایرکتوری باشد، خواهد توانست با تعیین سطوح دسترسی فایل یا دایرکتوری مربوطه برای دیگر کاربران یا گروه‌های کاربری، آنان را از امکان خواندن و ایجاد تغییر و یا اجرای فایل اجرایی محروم یا بهره‌مند سازد.

بدیهی است که مجوزهای تعریف شده برای یک گروه کاربری بر روی تمام اعضای آن گروه اعمال شده و هر کاربر عضو آن گروه از تمام مزایا یا محدودیت‌های موصوف بهره‌مند خواهد شد.

هسته‌ی لینوکس (Linux kernel) این امکان را به کاربرانی که هویت آنها تصدیق شده باشد می‌دهد تا بتوانند به فایل‌ها و دایرکتوری‌هایی که مجوز دسترسی دارند، دست یابند.

در لینوکس، هویت هر کاربر با یک شناسه‌ی کاربری که تحت‌عنوان uid (برگرفته از User ID) شناخته می‌شود، مشخص می‌گردد. در عین حال، یک بانک اطلاعاتی در سیستم‌عامل لینوکس وجود دارد که Username هر کاربر به‌همراه uid وی در آن به ثبت می‌رسند.

زمانی‌که در لینوکس یک کاربر جدید ایجاد می‌کنیم، اطلاعات این کاربر به بانک اطلاعاتی مذکور افزوده شده و در فایلی تحت‌عنوان passwd که در مسیر /etc/passwd قرار دارد، جدولی مانند جدول ۱ ایجاد می‌شود:

جدول ۱ | نمونه فایل passwd

نام فیلد	جزئیات	سایر توضیحات
Username	نام کاربری برای ورود به سیستم	می‌بایست بین ۱ تا ۳۲ کاراکتر طول داشته باشد.
Password	رمز عبور کاربر که با حرف x نمایش داده و رمزگذاری شده است.	رمز عبور زمانی که در لینوکس تایپ می‌شود به‌علت حفظ محرمانگی، هیچ‌گاه نمایش داده نمی‌شود.
User ID (UID)	شناسه‌ی گروه کاربری	این شناسه در مسیر /etc/group ذخیره می‌شود.
User Info	این فیلد اختیاری است و این امکان را به ما می‌دهد تا در صورت نیاز، اطلاعات بیشتری از کاربر ذخیره‌سازی کنیم.	-
Home Directory	مسیر قرارگیری دایرکتوری home کاربر	home/username/
Shell	مسیر قرارگیری Shell پیش‌فرض کاربر	برای مثال: /bin/bash

البته تنوع امکان ساخت user در سیستم‌عامل linux نیز می‌توانـد به حفظ امنیت اطلاعات کمک کند که در ادامه به معرفی انواع user در سیستم‌عامل لینوکس می‌پردازیم:

کاربر نوع اول: کاربر مدیر، root یا super user

زمانی‌کـه سیسـتم‌عامل لینوکـس را نصب می‌کنیـم کاربـر root یـا administrator به‌صورت خودکار ایجاد می‌شود.

این کاربر می‌تواند تمامی سرویس‌ها و جزئیات سیسـتم‌عامل لینوکس را مدیریـت کرده و به تمامـی منابع و بخش‌های سیسـتم‌عامل دسترسـی کامل دارد. مدیریـت نحوه‌ی کار سایر کاربران با سیستم (ایجاد/ حذف/ ویرایش کاربـران و همچنین تعیین و کنترل سـطوح دسترسـی کاربران به هر یـک از بخش‌های سیستم) بر عهده‌ی مدیر سیستم است.

دسترسـی‌های این کاربر آنقدر بالا اسـت که اشـتباهات احتمالـی وی می‌توانـد منجر به صدمه دیدن Kernel سیستم‌عامل شود و کل سیستم را دچار مشکل کند.

کاربر نـوع دوم: کاربـر عـادی، Regular یا Normal

دسـته‌ی بعدی، کاربران عـادی یا Regular Accounts هسـتند که این دسته از کاربران صرفـا می‌توانند بـه سیسـتم‌عامل لینوکس وارد شـده و کارهای روزمره و اسـتانداردی مثل اجرای برنامه‌های کاربردی، اسـتفاده از پردازشگرهای متنی (Text Editor)، استفاده از پایگاه‌هـای داده و مرورگرهـای وب و... را انجام دهند. در واقع کارهایی که به یک کاربر عادی داده می‌شود، نیاز به پیکربندی‌های فنی ندارد.

همان‌طور که ذکر شد، هر کدام از این کاربران برای خودشـان یک پوشه‌ی خانگی یا home directory دارنـد کـه می‌تواننـد درون این home directory فایل‌هـای خودشـان را ذخیره کنند. بـا توجه به اینکه کاربران عادی معمولا دارای دسترسی‌های مدیریتی نیستند، نمی‌توانند به‌صورت تصادفی فایل‌های حساس سیسـتم‌عامل را حذف یا دسـتکاری کرده و مشـکل خاصی برای سیسـتم ایجاد کنند. به این دسـته از کاربـران Normal Users هم گفته می‌شود.

کاربران نوع سـوم: کاربران سرویس Service Accounts یا Network Accounts

دسـته‌ی بعدی انواع کاربران سیسـتم‌عامل لینوکـس، کاربران سـرویس یـا Service Accounts هستند. همان‌طور که از اسم این نوع کاربران هم پیدا اسـت از این نوع حساب کاربـری در لینوکس بـرای ارائه‌ی خدمات یا سرویس استفاده می‌شود.

سـرویس‌هایی مثل وب سـرور آپاچی، cash server، ایمیل سـرورها، سرویس‌های DNS، بازی و حتی پرینت، بـرای انجام فرآیندهای خود در سیسـتم‌عامل هر کـدام به یک کاربر سـرویس نیاز دارند تا بتوانند اجرا شوند و به منابع موردنیاز خود در سیستم دسترسی پیدا کننـد. این نوع حسـاب‌های کاربری در واقع واسط بین سرویس‌های ما و کامپیوتر هستند تا سـرویس و کامپیوتر بتوانـد با هم صحبت کنند.

برخی از خدماتی که توسـط این سـرویس‌ها انجام می‌شود فقط برای سیستم‌عامل Local و برخی دیگر برای شبکه ارائه می‌شود و برخی از کاربـران Service Account می‌تواننـد فعالیت‌های شبکه را مانیتور و مدیریت کنند. به‌دلیل فعالیـت این کاربـران Service در شبکه از آنها به‌عنوان Network Account یا حساب کاربری شبکه‌ای هم یاد می‌شود.

کاربـر نوع چهارم: کاربر سیسـتم یـا System Account

بعـد از اینکـه سیسـتم‌عامل لینوکس نصب می‌شود، علاوه بر کاربر root که عنوان کردیم، یک نام کاربری از ما سئوال می‌شود تا بتوانیم بـرای کارهای روزمره‌ی خـود و قبل از ایجاد کردن هر نوع کاربر دیگر اسـتفاده کنیم. این کاربر نه دسترسی‌های کامل کاربر root را دارد و نه به اندازه‌ی کاربران عادی محدودیت دارد. در واقـع این کاربر بعد از مدیر سیسـتم، در سیستم‌عامل ایجاد شـده و می‌تواند به برخی از برنامه‌های امنیتی و تنظیمات سیستم‌عامل دسترسـی داشته و آنها را تغییر دهد. اگر یک کاربر عادی بخواهد ایـن کارها را انجام دهد، نیاز به دسترسـی‌های بالاتر و احراز هویت در سطح بالاتر را دارد.

کاربر نوع پنجم: کاربر Pseudo

این کاربر، در واقع یک نوع کاربر واسط است که ایـن امکان را می‌دهد تا بتوان در حال حاضر از دسترسی‌های کاربر بالاتر برای اجرای برنامه‌ها استفاده کرده و یا با دسترسی دیگری سرویس یا دستور خود را اجرا کرد. از Pseudo به‌عنوان یک کپی از کاربر root نام برده می‌شود.

در واقع در لینوکس این امکان فراهم شـده که در صـورت نیاز اجازه دهیم تـا کاربران عادی، بتوانند به‌صورت موقت از بعضی دسترسی‌های مدیر سیستم برخوردار شده و کارهایی که نیاز به سطح دسترسی ریشه دارند را عملی سازند. برای این کار، دو دستور مختلف وجود دارد که عبارتنـد از: sudo و su که تفاوت‌های این دو دستور در جدول ۲ ارائه شده است:

جدول ۲ | مقایسه‌ی دستورات sudo و su

su	sudo
زمانی‌که با استفاده از این دستور قصد داریم دسترسی‌های مدیریتی را به کاربری بدهیم، می‌بایست رمز عبور مدیر سیستم را وارد کنیم. (ارائه‌ی رمز عبور مدیر سیسـتم به کاربران نرمال هرگز توصیه نمی‌شود.)	زمانی‌کـه بـا اسـتفاده از این دسـتور قصد داریم دسترسی‌های مدیریتی را به کاربری بدهیم، نیاز به وارد کردن رمز عبور کاربر داریم نه رمـز عبور مدیر سیستم.
زمانی‌کـه کاربـری با اسـتفاده از دستور su به مجوزهای مدیریتی دسترسـی پیدا کند، ایـن کاربر امـکان انجام هـر کاری را خواهد داشت.	این‌دستور از قابلیت‌های بیشتری برخوردار بوده و به نوعی ایمن‌تر است.
ایـن دسـتور از قابلیت‌هـای اندکی برخوردار است.	این دسـتور قابلیت‌های زیادی دارا است.

۳- وجـــود قابلیت‌های فایل سیســـتم XFS در لینوکس

یکی دیگـــر از مزایای انتخاب شـــبکه‌ی تحت لینوکس برای سایت‌های نفتی، امکان استفاده از فایل سیستم xfs در سیستم‌عامل لینوکس، در این‌گونه شبکه‌های مهم است.

این ماهیت در سیستم‌عامل، نحوه‌ی نگهداری و ســـاختار نگهداری فایل‌ها بر روی پارتیشن‌ها و دیسک‌ها را مشـــخص می‌کند. فایل سیستم XFS از انواع فایل سیستم‌های جدید محسوب شده و برای استفاده در محیط‌هایی که با حجم زیادی فایل سروکار دارند به‌صورت ویژه طراحی شده است.

در واقع Xfs یکی از فایل سیســـتم‌هایی است که به‌عنوان فایل سیستم جانبی برای سرورهای لینوکسی می‌تواند جایگزین ext۳ و ext۴ شود و در محیط‌های کلان ســـازمانی کاربرد خوبی دارد.

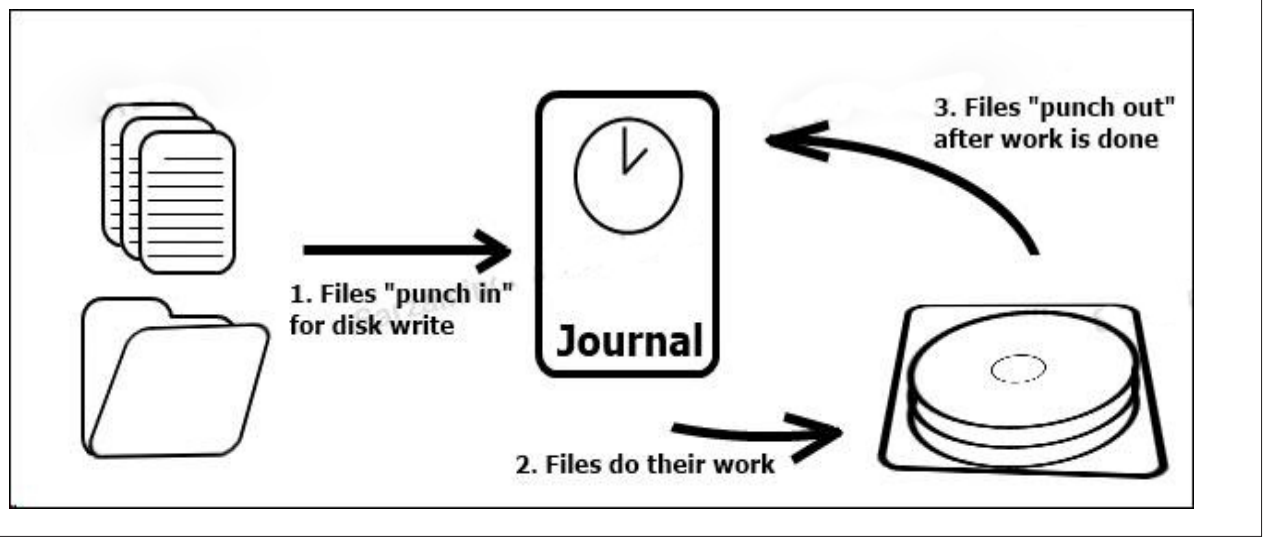
در ایـــن گزارش به بیان بخشـــی از قابلیت‌های فایل سیستم xfs و مزایای این فایل سیستم در شبکه‌های نرم‌افزاری تحت‌لینوکس می‌پردازیم.

فایل سیســـتم xfs به‌عنوان یک فایل سیستم دارای قابلیت Journaling است که با استفاده از الگوریتـــم تعادل B-Tree طراحی شـــده و امکان پیدا کردن فایل‌ها را در سریع‌ترین زمان ممکن به ما می‌دهد. یکی از اولویت‌هایی که در طراحی این فایل سیســـتم در نظر گرفته شده اســـت، پشـــتیبانی از فایل‌های بسیار حجیم و تعداد فایل‌های بسیار زیاد است.

پیـــش از پرداختن به معماری ســـاختار xfs و مزایای این فایل سیستم به توضیحی کوتاه در مورد Journaling می‌پردازیم:

در شرایطی که در حال انتقال data، به هر دلیلی از جملـــه قطع ناگهانی برق یا قطعی سیســـتم مواجه شـــویم (که در این‌صورت امکان تخریب و حذف اطلاعـــات، بالا اســـت)، وجود ویژگی Journaling در فایل سیستم از تخریب دیتا جلوگیری به‌عمل می‌آورد. در شکل۵ کلیات این موضوع نمایش داده شده است.

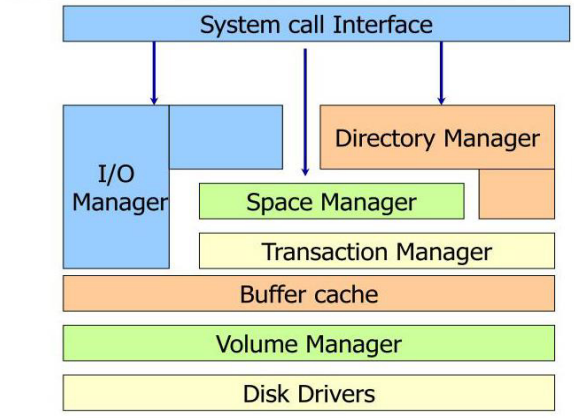
هنگامی‌که کامپیوتر به‌طـــور ناگهانی خاموش می‌شـــود، ممکن اســـت اطلاعـــات حافظه‌ی ذخیره‌سازی دائمی دچار آسیب شوند. چون در این حالت مقداری از اطلاعات بر روی حافظه‌ی ذخیره‌ســـازی موقت و در حال پردازش توسط پردازشگر است و بخش دیگری روی حافظه‌ی ذخیره‌ســـازی دائمی قرار دارد و سیستم فایل وارد حالتی به‌نـــام Half-Finished یا حالت نیمه‌تمام می‌شـــود. معمولا در این‌گونه موارد، سیستم پس از روشن شدن مجدد، روال خاصی را جهت بررسی اشکالات حافظه‌ی دائمی دنبال می‌کند. این امر ممکن است زمان زیادی به‌طول بینجامد و بـــرای اطمینـــان از صحت کارکرد سیســـتم فایل بایستی در هر بار شروع سیستم این روال دنبال شود.



شکل ۵ | استفاده از روش Journaling در فایل سیستم

همان‌گونه که در شـــکل۵ مشاهده می‌شود، در این‌گونـــه فایل سیســـتم، قســـمتی به‌نام Journal وجـــود دارد که تمـــام تغییراتی را که باید در storage ثبت شـــوند را همانند یک سیســـتم گزارش‌گیری، ذخیره می‌کند. به ایـــن مکانیزم در اصطلاح، سیســـتم ثبت وقایـــع فایل‌هـــا می‌گوینـــد. (یک پروســـه در Background به‌صـــورت نامحســـوس تغییـــرات را دنبال کرده و اطلاعات موردنیاز را در Journal می‌نویسد). این موضوع باعث می‌شـــود تا در صورت بروز هر گونه مشکل، امکان رفع نقص ســـریع‌تر شـــده و از طرفی نیاز به بررســـی سیســـتم فایل در هر بار که سیستم شروع به‌کار می‌کند، نباشد.

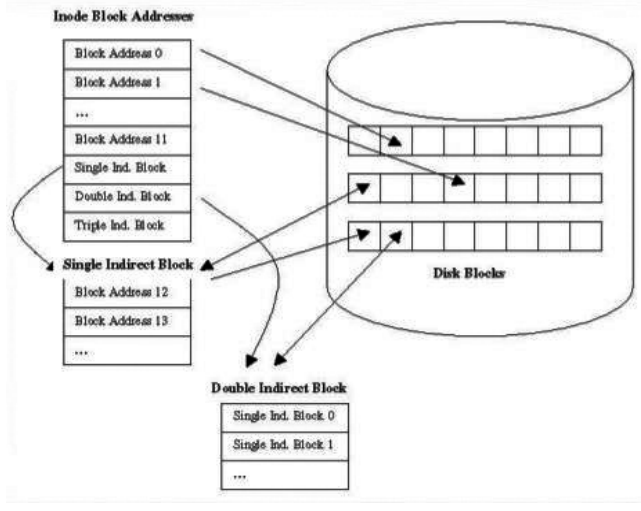
اینک به توضیح بیشتر نحوه‌ی عملکرد فایل سیســـتم xfs می‌پردازیم. معماری کلی فایل سیســـتم مذکور در شکل۶ نمایش داده شده است.



شکل ۶| معماری کلی فایل سیستم xfs

خیلی از وظایف مرتبط با فایل سیستم در xfs به شکل متفاوتی انجام می‌شوند که از جمله آنها می‌توان به quota ، تعمیر فایل سیستم و حتی مکانیزم کپی کردن فایل‌ها اشاره کرد. این تفاوت‌ها به خاطر نحوه‌ی عملکرد متفاوت فایل سیستم xfs در مقایسه با سایر فایل سیستم‌ها است. یکی از مهم‌ترین قابلیت‌های xfs که باعث می‌شود ما بتوانیم فایل‌های بسیار حجیم و تعداد زیادی فایل را نگهداری کنیم قابلیتی به‌نام allocation group است که مقیاس‌پذیری (Scalability) بسیار زیادی به این فایل سیستم می‌دهد. هر یک از allocation group هایی که در xfs وجود دارند، برای خودشان inode های خاص دارند و فضای خالی و ایجادallocation group های دیگر را مدیریت می‌کنند که با این مکانیزم امکان استفاده‌ی همزمان از فایل سیستم برای پردازش‌ها و thread ها برای دسترسی به سرویس‌ها فراهم می‌آید. این موضوع به شدت در محیط‌های Enterprise باعث بالا رفتن کارآیی سیستم می‌شود.

یکی از دلایل کارآمدی xfs ، نحوه‌ی مدیریت extent ها است. در واقع یک extent مجموعه یا گروهی از block ها است که در قالب یک موجودیت، آدرس‌دهی می‌شوند. مزیت استفاده از extent ها این است که به‌جای اینکه تک‌تک فایل‌های موجود بر روی سیستم آدرس‌دهی شوند (تک‌تک بلوک‌های حافظه) فایل سیستم تنها تعداد بسیار کمتری extent را آدرس‌دهی می‌کند و در نتیجه در آدرس‌دهی تعداد فایل‌های بسیار زیاد عملکرد و کارآیی بسیار بهتری را ارائه می‌دهد. شکل۷ به‌صورت شماتیک این موضوع را نشان می‌دهد.



شکل ۷| آدرس‌دهی در xfs

یکی دیگـر از قابلیت‌های بسـیار مفید xfs قابلیتـی به‌نـام delayed allocation یـا تخصیص فضای با تاخیر است. زمانی‌که یک فایل ایجاد می‌شـود، محتویات آن در حالت عادی در بافر cash و سپس بر روی دیسک نوشـته می‌شـوند. Xfs در ایـن فاصله‌ی زمانی تا جایی‌که ممکن اسـت تاخیر ایجاد می‌کنـد زیرا یک فایل جدیـد قبل از اینکه بر روی دیسک نوشته شـود یا بعد از ایجاد آن اطلاعاتـی دارد کـه آن را تغییر می‌دهد و تاخیـر موجود این فرصـت را می‌دهد که تغییرات قبل از نوشته شدن بر روی دیسک سریع‌تر نوشـته و ذخیره شـوند. با داشتن کمی تاخیر در نوشتن فایل بر روی دیسک، سیسـتم این فرصـت را پیـدا می‌کند که بلوک‌های حافظه‌ی نزدیک و پشت‌سـرهم را پیدا کرده و فایل را برای دسترسـی سریع‌تر آماده کند. در واقع Delayed Allocation تا حـدود زیادی جلـوی Fragmentation داده‌ها بر روی دیسک را می‌گیرد.

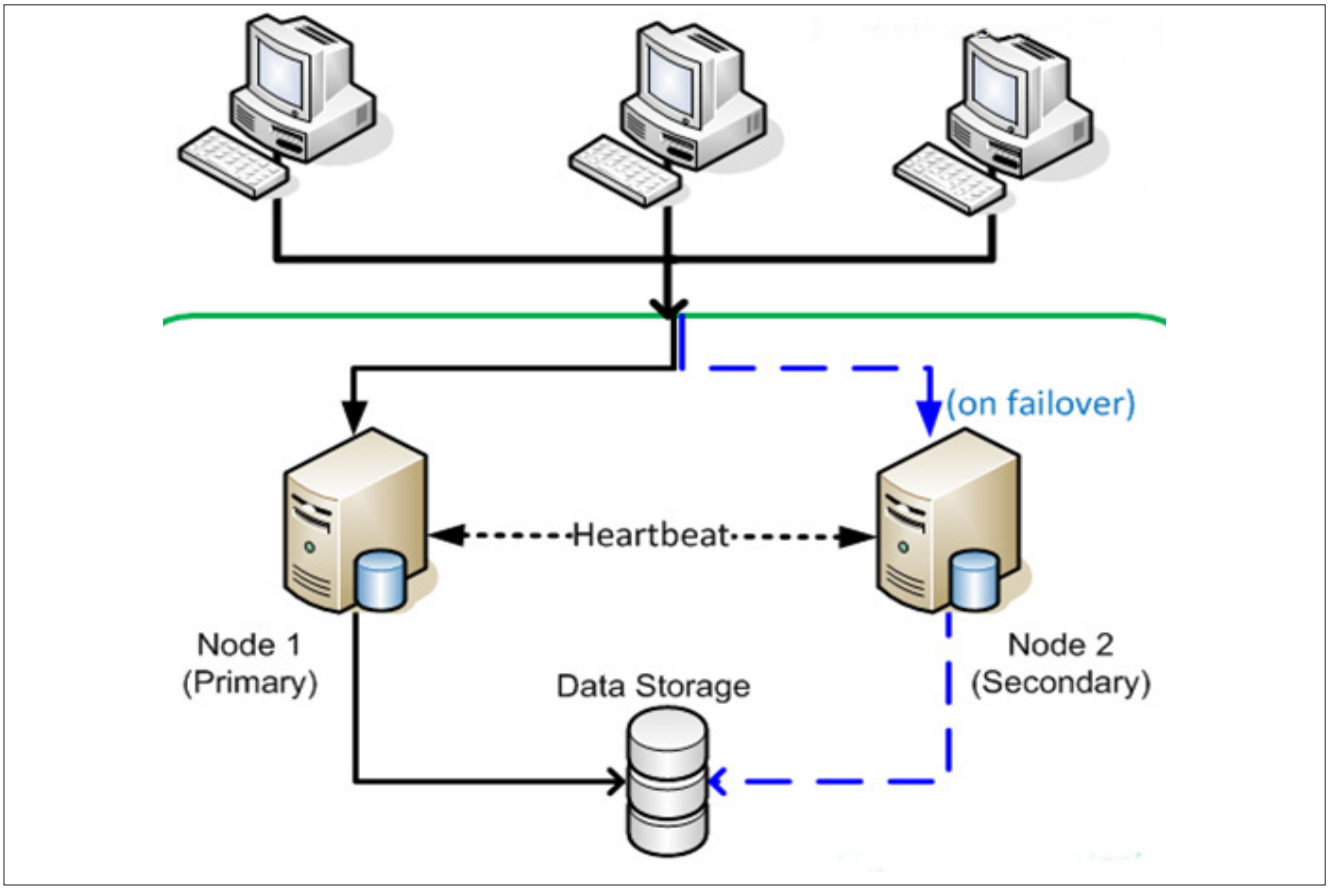
قابلیت دیگری که در xfs وجـود دارد Direct I/O اسـت که فایـل به‌هیچ عنوان بافر نشـده و مسـتقیما بر روی دیسک نوشـته می‌شود. این‌کار باعـث کاهش میزان I/O سیسـتم می‌شود و xfs این فرآیند را گارانتی می‌کند.

۴- امکان ایجاد Clustering

از دیگر مزایای انتخاب شبکه‌ی تحت‌لینوکس برای سـایت‌های نفتی، پشـتیبانی بسـیار خوب لینوکس از تکنیک کلاسترینگ است. همان‌گونه که می‌دانیم در سایت‌های مذکور از نرم‌افزارهـای تخصصـی نفتی اسـتفاده می‌شود. اطلاعات تولید شده (data)ها بسیار پرحجم بـوده و بعضا عملیات و محاسـبات پیچیـده و زمان‌بری روی اطلاعـات انجام می‌شود.

بـه دلایل مذکور، کلاسـترینگ در این‌گونه شـبکه‌ها بسـیار مفید و در بعضی شرایط، ضروری است.

کلاسترینگ در لغت به‌معنای "خوشه‌بندی" بوده و مفهوم آن در شـبکه این اسـت که چندین سـرور به‌طور همزمان یک سرویس را ارائـه می‌دهنـد و هر سـرور را می‌توان به‌عنوان یک خوشـه در نظـر گرفت. مفهوم کلاسـترینگ، به‌طـور کلـی در شـکل۸ نمایش داده شـده اسـت. البته به‌کارگیری و پیاده‌سـازی این مفهوم در شـبکه مزایای بسـیاری دارد که در ادامـه‌ی مطلب به آن پرداخته خواهد شد.



شکل ۸ | مفهوم کلاسترینگ

یکی از مزایـای به‌کارگیری "کلاسـترینگ" افزایش سـرعت عملکرد در ارائه‌ی سـرویس اسـت. دلیـل این امر این اسـت کـه وقتی درخواستی به مجموعه‌ی کلاستر شده می‌رسد چنانچه اولین سـرور نتواند به درخواست فوق پاسـخ دهد درخواسـت فوق به سرور بعدی ارجاع داده می‌شود (Failover Clustering) و چنانچه همه‌ی سرورها بدون هیچ مشکلی به درخواسـت‌ها پاسـخ دهند بـار پردازش درخواست‌ها بین سرورها توزیع شده و سبب Load Balancing می‌شود که در نهایت به افزایش سرعت پاسـخگویی به درخواست‌ها منجر می‌شود.

سـرویس کلاسـترینگ برای برنامه‌هایی که تحت هر شرایطی باید اجرا شده و در حالت اجرا باقـی بمانند (حتی زمانـی که یکی از سرورها از سـرویس خارج شده باشد) بسیار مناسب است. سرورهای کلاستر بیشتر برای برنامه‌هایـی به‌کار می‌روند کـه مدت زمان زیـادی را در حافظه می‌ماننـد و یا تعویض داده‌ی بیشتری را انجام می‌دهند.

مفهوم کلاسـترینگ یا همان خوشه‌بندی در محیط یـک Application Server زمانی مطرح می‌شـود که بخواهیم سیسـتم‌هایی مقیاس‌پذیر و مقاوم در برابر خرابی داشـته باشـیم به این معنـی که با افزایـش بار و بالا رفتـن تعداد درخواسـت‌ها از سـرور، سـرور قدرت پاسـخ‌دهی به حجم بیشتری درخواست را داشته باشد و یا در صورت بروز خرابی سامانه همچنان قادر به پاسخگویی و تحمل شرایط خرابی باشد.

بـرای توضیح بیشـتر مقیاس‌پذیری، که در شـبکه‌های موضوع بحث مـا، نقش مهمی را ایفا می‌کند، مبحث انـواع مقیاس‌پذیری مطرح می‌شود.

در کل، مفهـوم مقیاس‌پذیری به دو شـکل: عمودی و افقی مطرح می‌شود.

در مقیاس‌پذیری عمودی ما سـرور موجود را بـه تجهیـزات سـخت‌افزاری بیشـتر و قوی‌تر مجهـز می‌کنیم. به‌عنوان مثال تعداد بیشـتری حافظه به سـرور اضافه می‌کنیم یا دیسـک سخت بیشـتری در اختیار سرور قـرار می‌دهیم. گاهی نیز تعداد هسـته‌های پردازنـده را افزایش می‌دهیم. حُسـن این روش در این است که سیستم‌عامل به‌صورت اتوماتیک، سـخت‌افزار جدیـدی که افزوده شـده اسـت را تشـخیص می‌دهد و تقریبا نیازی به تغییر در سطح برنامه‌های کاربردی نیست.

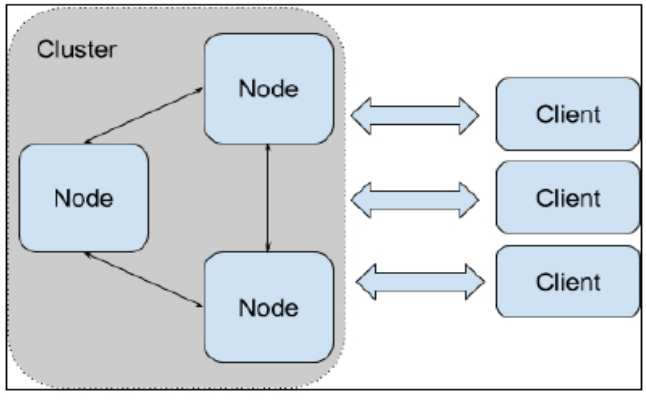
در مقیاس‌پذیری افقی، ما سـرورهای جدید به مجموعه‌ی سـرور موجود اضافه می‌کنیم به این معنا که سرور قدیمی موجود تغییری نخواهد داشت بلکه ما با شبکه‌ای از سرورها (که به اصلاح به هر سرور در این شبکه گره یا نود گفته می‌شـود) شـروع به‌کار کرده و به مرور این شـبکه را گسترش می‌دهیم. در حقیقـت به مرور ما به این شـبکه، گره‌های بیشـتری می‌افزاییم. در مقیاس‌پذیری افقی معمولا نیاز به پیکربنـدی و تنظیم (tune) نودهای جدید و باز نویسـی application داریم.

در ادامه به توضیح مختصری در مورد مفاهیم مهم موجود در کلاسترینگ می‌پردازیم:

■ خوشه (cluster):

خوشـه یا کلاسـتر به مجموعه‌ای از نودهای کامپیوتری توزیع شـده گفته می‌شـود که با یکدیگر شـبکه شـده‌اند. معمولا این شـبکه یک شبکه‌ی LAN اسـت. هدف این نودهای کامپیوتـری تقسـیم بهینـه‌ی کار موجود، از طریق ارتباط با یکدیگر اسـت. از زاویه‌ی دید کسی که از خوشه استفاده می‌کند خوشه مانند یک سیسـتم واحد و یک کامپیوتر واحد دیده می‌شود.(شکل۹)

در حقیقت هدف اصلی از تشـکیل کلاسـتر، توانایی پاسـخ به تعداد بیشـتری درخواست و انجام حجم بیشـتری از کار در زمان کمتر، در مقایسه با یک ماشین واحد است که این موضوع در شـبکه‌های نرم‌افزاری تخصصی (مخصوصا زمانی که نیاز به محاسـبات ریاضی پیچیده و پروسـه‌های زمان‌بر دارند)، مزیت بسیار مهمی محسوب می‌شود.



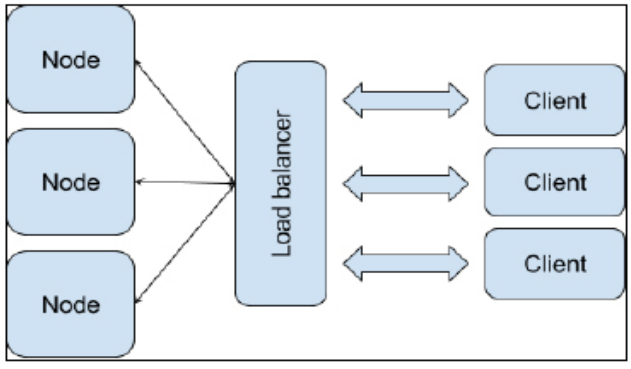
شکل ۹ | عملکرد خوشه به‌عنوان یک سیستم واحد

در سایت‌های نرم‌افزارهای تخصصی، تشکیل خوشه هم به‌معنی افزایش تعداد application server ها بر روی یک نود (یعنی یک سرور) و هم به‌معنی افزایش تعداد application server ها به‌همراه نودهای جدید است.

■ متوازن‌ساز بار (load balancer):

معمولا هر نود در خوشه قابلیت توازن بار را دارد به این معنی که در صورتی که بار کاری یک نود بسیار بالا است، آن نود درخواست‌های جدید را به نود دیگر در همان خوشه واگذار می‌کند. اما گاهی این وظیفه به عهده‌ی مولفه‌ای دیگر به اسم متوازن‌ساز بار یا همان Load Balancer گذاشته می‌شود به‌نحوی که هر درخواست ورودی به خوشه ابتدا توسط متوازن‌ساز بار بررسی و سپس با استفاده از یک الگوریتم مشخص، به نودی در خوشه واگذار می‌شود که برای پردازش درخواست متناسب‌تر باشد. معمولا نودی انتخاب می‌شود که دارای حجم باری کمتری است.

بدین‌ترتیب درخواست‌های جدید به سمت نودهای با فشار کاری بالا هدایت نمی‌شوند. در صورتی که بار کاری یک نود که قبلا تحت فشار کاری بالا بوده است، کم شود مجددا متوازن‌ساز بار درخواست‌های جدید را به‌سمت این نود هدایت می‌کند. این موضوع در شکل ۱۰ نشان داده شده است. در حقیقت متوازن‌ساز بار به نوعی تقسیم‌کننده‌ی درخواست‌ها بین نودهای موجود است به نحوی که به هر نود بیش از حد توان، بار و فشار تحمیل نشود.



شکل ۱۰ | استفاده از متوازن‌ساز بار (load balancer) در کلاستر

روش‌های مختلفی برای تقسیم بار وجود دارد که توسط متوازن‌ساز بار به‌کار گرفته می‌شوند. بعضی از آنها عبارتند از:

۱- ارسال درخواست ورودی جدید به اولین نود موجود و در دسترس

۲- ارسال درخواست ورودی جدید به‌صورت نوبتی

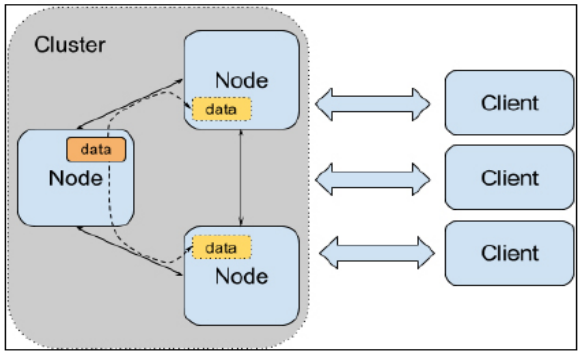
۳- روش sticky session که در آن درخواست‌های بعدی در یک session همواره به همان نودی هدایت می‌شوند که اولین درخواست این session به آن فرستاده شده بود.

نکته‌ی مهم این که متوازن‌ساز بار به پارامترهای مختلف یک نود نگاه می‌کند، مثلا به میزان cpu usage آن نود و یا پهنای باند آزاد یک نود و در صورت مناسب بودن شرایط و کم بودن بار کاری، درخواست را به آن نود هدایت می‌کند.

■ رونوشت (replication):

معمولا در هر یک از نودهای یک خوشه (هر یک از سرورها) یک رونوشت از داده‌ها و محاسبات انجام شده در سطح خوشه (در سرورهای دیگر) وجود دارد.(شکل ۱۱)

در حقیقت، وجود رونوشت امکان به اشتراک‌گذاری داده در سطح خوشه را در اختیار نودها قرار می‌دهد و جامعیت داده را برای خوشه فراهم می‌کند.■



شکل ۱۱ | رونوشت یا replication در کلاسترینگ

پانویس‌ها

۱. شماره‌ی ۱۷۷ ماهنامه‌ی اکتشاف و تولید.