آشنایی با مفهوم کاسترینگ در مهندسی کامپیوتر و لینوکس

معنی لغوی کلاسترینگ یعنی خوشه بندی ...کلاسترینگ عبارت است از کار کردن چند سیستم به عنوان یک سیستم واحد به طوریکه بازدهی یک سیستم چند برابر میشود.... کلاسترینگ در زمینه های مختلف با اهداف خاصی مورد استفاده قرار میگیره... در کامپیوتر بیشتر برای تقسیم بار و کم کردن درصد خطا مورد به کار می‌رود... کلاستر از دید سیستم عامل به سیستم های توضیع شده ربط داده می‌شود که بتوان با منابع مختلف به جای یک سیستم از دید کاربر نشان دهیم...

يک مجموعه کامپيوتر کلاستر گروهي از کامپيوترهاي آزاد است که با هم کار مي کنند و از بسياري جهات مي توان به مجموعه رفتار آن ها به عنوان يک کامپيوتر نگريست. اجزاي تشکيل دهنده اين کامپيوتر کلاستر معمولا از طريق شبکه به همديگر متصلند. کلاسترها معمولا براي بالا بردن سرعت، ايجاد افزونگي براي افزايش قابليت دسترسي مداوم و امن کردن موقعيت در هنگام خطا و ايجاد Load Balancing پيکربندي مي‌شوند .

مثلا برای cache server به این صورت است که صفحات در هارد سرورهای مختلف دخیره می شوند و درخواست کلاینت به نزدیکترین سرور می رسد حال اگر این سرور مشغول بود این درخواست به سرور بعدی می رود یا اصلا ممکن است سایت مورد نظر در آن سرور نباشد به این ترتیب سرور اول کلاینت را به سروری هدایت می کند که حاوی این صفحه باشد بدین ترتیب هم حجم هارد را چند برابر کرده ایم و هم سرعت بازیابی صفحه چند برابر شده است.

- بحث کلاسترینگ بر روی سرورها به این مفهوم است که گروهی از سرورها باهم یک برنامه خاصی رو اجرا می‌کنند که هر سرور به عنوان یک خوشه بخشی از کار رو انجام میده که نتیجه آن سرعت بالای اجرا می باشد زیرا بار محاسباتی بین این سرورها توزیع میشه که اصطلاحا بهش load balancing میگن علاوه بر این مزیت دیگه ای که در حضور چند سرور و کلاسترینگ وجود داره اینه که اگر یه سرور از کار بیفته دیگر سرورها باید به درخواستها پاسخ بدن که به این عمل failover گفته میشه. هنگامی که سروری از سیستم خارج می‌شود، Failover اتفاق می‌افتد و با بکار افتادن دوباره‌ی آن(Failback) ، بقیه سرورها اطلاع پیدا کرده و روند عادی کار دوباره ادامه خواهد یافت. اکثر وب سرورها از این ساختار استفاده می‌کنند چون تعداد requst های آنها بالاست.

سرویس کلاسترینگ بیشتر برای سازمان و شرکتهایی به کار میرود که برنامه ی آنها تحت هر شرایطی باید اجرا شود و در حالت اجرا باقی بماند حتی زمانی که یکی از سرورها از سرویس خارج شده باشد. سرورهای کلاستر بیشتر برای برنامه هایی به کار میروند که مدت زمان زیادی را در حافظه میمانند و یا تعویض داده ی بیشتری را انجام میدهند که به آنها Stateful Applications گفته می‌شود که می‌توان سرورهای بانک اطلاعاتی نظیر Microsoft SQL یا سرورهای ایمیل مایکروسافت را نام برد.

(HPC) High Performance/Productivity Computing

سرويس HPC (High Performance/Productivity Computing) بر روي Microsoft Windows يا سيستم عامل هاي linux راه اندازي مي شود. بيشترين كاربرد اين تكنولوژي براي حل مسائل علمي مي باشد. به همين دليل گاهي اين تكنولوژي را Supercomputing خطاب مي كنند. راه اندازي بستر HPC زمينه مناسب را براي Run گرفتن از نرم افزارهاي مهندسي كه اين بستر را مي شناسند و Support مي كنند فراهم مي كند تا مسائل بزرگ را در مدت زمان كوتاه تر و با هزينه خريد و راه اندازي مناسب تر از Supercomputers آناليز و حل نمود.

کلاسترها در لینوکس

با رشد انفجاری اینترنت، کار بر روی سرورهای وب، ایمیل و مدیا افزایش می‌یابد. سایت‌های زیادی در حال مبارزه برای حفظ کردن خود در برابر تقاضاهای رو به رشد هستند و تکنیک‌های زیادی را برای جلوگیری از overload شدن سرورهای خود به کار می‌گیرند. راه‌اندازی یک سرور بر روی یک گروه کامپیوتر (cluster) یکی از راه‌هایی هست که بطور موثر مورد استفاده قرار می‌گیرد. با این روش در صورت لزوم می‌توان با اضافه کردن یک سرور جدید به گروه سرورهای موجود، تقاضاهای بیشتر را به راحتی مدیریت کرد.

مبحث کلاسترها در لینوکس یکی از جذاب‌ترین و جالب‌ترین مباحث برای افراد علاقه‌مند به پردازش‌های موازی است .

کلاسترها چه هستند؟

به طور عمومی هنگامی که صحبت از کلاسترها می‌شود، مقصود فناوری‌هایی است که از طریق آن کامپیوترهای مختلف بتوانند با هم و با اشتراک قدرت پردازش هم، بتوانند امور پردازشی را که به آنها محول شده است، انجام دهند. این امور پردازشی همه چیز می‌تواند باشد. از پردازش‌های سنگین علمی تا تبدیل فایل‌های موسیقی و یا رندر کردن جلوه‌های ویژه فیلم‌های سینمایی. برای مثال، تمامی جلوه‌های ویژه فیلم‌های ارباب

حلقه‌ها توسط کلاسترهای لینوکس رندر و پردازش شده‌اند.

تواع مختلفی از فناوری‌های کلاستر سازی برای سیستم‌عامل لینوکس وجود دارند. یکی از شناخته شده ترین آنها کلاستر Beowulf است. این کلاستر حاوی چندین ماشین است که توسط یک شبکه محلی پرسرعت به هم متصل شده‌اند. برای استفاده از این سیستم‌های کلاستر، برنامه‌های کاربردی باید مجددا برای استفاده از آن با استفاده از کتابخانه‌های کلاستر سازی نوشته شوند. عمومی‌ترین کتابخانه‌های کلاستر سازی عبارتند از PVMو .MPI هر دوی این کتابخانه‌ها بسیار عالی کار می‌کنند. با استفاده این کتابخانه‌ها، برنامه نویسان قادر به نوشتن برنامه‌هایی هستند که از منابع روی کلاستر همانند منابع روی یک کامپیوتر، بهره گیری نمایند.برای بسیاری از برنامه‌های کاربردی، PVMو MPI امکان افزایش خطی قدرت پردازش کلاسترها را با توجه به تعداد ماشین‌های روی آن فراهم می‌نمایند.

PVMو MPI به درد همه نمی‌خورد!

با اینکه کلاسترهای Beowulf بسیار قدرتمند هستند، ولی به درد همه کس نمی‌خورند! بزرگترین اشکال آنها نیاز به نرم‌افزارهای خاص می‌باشد که با استفاده از PVM و MPI نوشته شده باشند تا بتوانند از مزایای کلاستر استفاده کنند. البته این برای مراکز علمی و تحقیقاتی که برنامه‌های کاربردی خاص خود را از ابتدا می‌نویسند، اشکال مهمی نیست. آنها به راحتی قادرند تا از MPI و PVM استفاده کنند.

حقیقتا درصد افراد و موسساتی که برنامه‌های کاربردی خود را از ابتدا می‌نویسند بسیار پایین است. برای کسانی که مایل هستند تا یک کلاستر بنا کرده و از مزایای آن در اجرای برنامه‌های کاربردی عادی استفاده کنند، این یک مسئله بزرگ است! برنامه‌های کاربردی این دسته از موسسات بدون استفاده از کتابخانه‌های کلاستر سازی نوشته شده‌اند، بنابراین این گونه موسسات قادر نیستند تا از مزایای کلاسترها بهره‌گیری نمایند.

آیا جالب نیست که یک فناوری وجود داشته باشد تا بتوانید با استفاده از آن از مزایای کلاسترهای لینوکس استفاده کنید، بدون آنکه نیاز داشته باشید تا برنامه‌های کاربردی خود را از ابتدا نوشته و یا حتی آنها را مجددا کامپایل نمایید؟ خوشبختانه چنین فناوری وجود دارد و نام آن OpenMosix است!

ورود به OpenMosix

OpenMosix قابلیت‌های کلاستر سازی را به هسته لینوکس اضافه می‌کند، بنابراین هر پروسه استاندارد لینوکس قادر خواهد بود تا از مزایای منابع کلاستر استفاده نماید. با استفاده از تکنیک‌های موازنه بار تطبیقی (Adaptive Load Balancing)پردازش‌های در حال اجرا بر روی یک گره (node) از کلاستر، قادرند تا بطور نامحسوس به یک گره دیگر از کلاستر مهاجرت کرده و بتوانند سریعتر اجرا شوند. بدلیل اینکه OpenMosix بطور کاملا نامحسوس (Transparent) عمل می‌کند، پردازش‌هایی که از یک گره به گره دیگر مهاجرت می‌کنند، حتی نمی‌دانند (لازم هم نیست بدانند) که در یک ماشین دیگر در حال اجرا هستند!
نامحسوس بودن OpenMosix به این معنی است که برای استفاده از مزایای موازنه بار تطبیقی آن، نیازی به برنامه نویسی خاصی نیست. در حقیقت، یک نصب پیش‌گزیده OpenMosix به طور خودکار پردازش‌ها را به بهترین گره منتقل خواهد کرد. این قابلیت OpenMosix را تبدیل به یک راه‌حل کلاستر سازی می‌کند که می‌تواند برای بخش عظیمی از برنامه‌ها مفید باشد.

OpenMosix دقیقا چکار می‌کند؟

بزرگترین کاری که OpenMosix انجام می‌دهد، تبدیل دسته‌ای از ماشین‌های لینوکس به یک سیستم بزرگ مجازی چند پردازنده‌ای متقارن (SMP=Symmetric MultiProcessor) است. هرچند نحوه عملکرد آن با سیستم‌های SMP واقعی مقداری تفاوت دارد. نخست اینکه سیستم‌های واقعی SMP که مبتنی بر ۲ یا چند پردازنده هستند، می‌توانند اطلاعات را با سرعت بسیار بالا تبادل نمایند، در صورتی که در OpenMosix سرعت

ارتباط بین گره‌های کلاستر، محدود به سرعت شبکه محلی است که گره‌ها در آن قرار دارند. استفاده از ارتباطات اترنت گیگابیت و یا سایر انواع پر سرعت اترنت باعث خواهد شد تا تبادل داده‌ها با سرعت بالاتری صورت گرفته و کارایی کلاستر بالاتر باشد.

البته OpenMosix دارای مزایایی نسبت به سیستم‌های چند پردازنده‌ای سنتی داراست. با استفاده از OpenMosixشما قادر به ایجاد کلاسترهایی حاوی دها و حتی صدها کامپیوتر با سخت‌افزار ارزان هستید در حالی که سیستم‌های SMP که حاوی تعداد زیادی پردازنده باشند، می‌توانند بسیار گرانقیمت باشند. برای بسیاری از برنامه‌های کاربردی، OpenMosix نسبت به سیستم‌های SMP یاMainframe ، حرف بیشتری برای گفتن دارد. البته دلیلی وجود ندارد که شما نتوانید OpenMosix را بر روی سیستم‌های قدرتمند چند پردازنده‌ای اجرا نمایید. حتی این امکان وجود دارد تا OpenMosix را به همراه برنامه‌های کاربردی که با MPI یا PVM توسعه یافته‌اند، اجرا نمایید تا سرعت کلاستر خود را بهینه نمایید.

همانند سیستم‌های SMP سنتی، OpenMosix قادر نیست تا یک پروسه را روی چند پردازنده فیزیکی اجرا نماید. واضح‌تر اینکه نباید انتظار داشته باشید تا اجرای برنامه‌ای مانند مرورگر موزیلا روی یک کلاستر سریعتر از یک سیستم تک پردازنده‌ای باشد، مگر اینکه اجرا پروسه آنرا به یک گره سریعتر روی کلاستر منتقل نمایید.

بعلاوه در حال حاضر OpenMosix امکان جداسازی رشته‌های متعدد به هم پیوسته را از یکدیگر فراهم نمی‌کند. OpenMosix قادر است تا پروسه‌های استاندارد لینوکس را بین گره‌های کلاستر بدون مشکل مهاجرت دهد. در صورتی که یک برنامه کاربردی تعداد زیادی زیر پروسه داشته باشد، آنگاه OpenMosix قادر است تا هر یک از آنها را به یک گره مناسب در کلاستر منتقل کند. شما می‌توانید از این قابلیت حتی در برنامه‌های کاربردی که دارای زیر پروسه نیستند نیز استفاده کنید. برای مثال، در صورتی که نیاز دارید تا تعدادی فایل موسیقی را از فرمت mp3به wav تبدیل نمایید، تبدیل هر فایل یک پروسه خواهد بود. شما می‌توانید تمام این پروسه‌ها را یکجا اجرا نمایید. در آنصورت عمل پردازش بین کلاستر پخش خواهد شد (بجای اینکه عملیات تبدیل فایل‌ها را یک به یک انجام دهید). در صورتی که شما ۱۲ فایل موسیقی و ۱۲ گره همسان داشته باشید، عملیات تبدیل ۱۲ بار سریعتر انجام خواهد شد.