شاید خیلی از شما کاربران عزیز با اصطلاح کلاسترینگ آشنایی نداشته باشید. خیلی از شما حتی نیازی هم ندارید که بدانید کلاسترینگ به چه معناست. اما کسانی که کمی تخصصی تر به قضیه نگاه کنند و بحث های شبکه و networking رو دنبال کنند مطمئناً میدونند کلاسترینگ از حیاتی ترین و مهمترین تکنولوژی هاییست که هر شرکت بزرگی باید تابع و استفاده کننده از این تکنولوژی باشه در غیر این صورت یک بازنده در دنیای نت به حساب میاد. شرکت های بزرگی همچون گوگل، یاهو، مایکروسافت و خیلی از شرکتهای دیگه از کلاسترینگ برای ثبات و دوام سرورهای خودشون و سرویس دهی به کاربران استفاده میکنند.

در ادامه به معرفی کلاسترینگ و اینکه چرا هر شرکتی و سرویس دهنده ای در دنیای نت **باید** از این تکنولوژی استفاده کند، خواهیم پرداخت.

مرکز تحقیقات امن داده های پایدار (پادکو)
PADCO - Datacenter

کلاسترینگ چیست؟

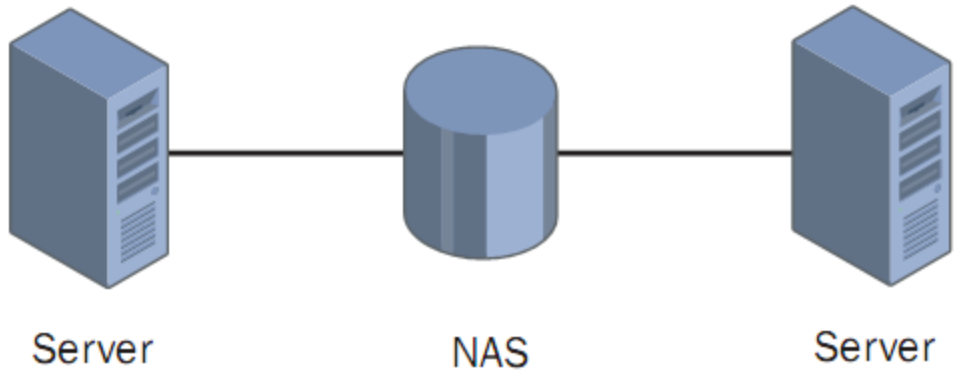
کلاسترینگ به گروهی از سرورها شامل یک یا بیشتر از یک سرور گفته میشود که با یکدیگر نرم افزاری مشخص را اجرا میکنند که نتیجه ی آن احتمال خطای کمتر و سرعت لد بالاتر برنامه هاست. سرویس کلاسترینگ بیشتر برای سازمان و شرکتهایی به کار میرود که برنامه ی آنها تحت هر شرایطی باید اجرا شود و در حالت اجرا باقی بماند حتی زمانی که یکی از سرورها از سرویس خارج شده باشد. در سرویس کلاسترینگ تمامی سرورها همان برنامه و قوانینی را که باقی سرورها اجرا میکنند اجرا خواهند کرد لذا اگر سروری از سرویس خارج شود باقی سرورها بلافاصله شروع به کار خواهند کرد که به آن Failover گفته میشود. زمانی که سرور از سرویس خارج شده دوباره به کار افتاد باقی سرورها آگاه میشوند و روند عادی دوباره ادامه خواهد یافت. به این عمل Failback گفته میشود.

ویندوز سرور ۲۰۰۳ دو نوع کلاسترینگ را پشتیبانی میکند:

۱. سرورهای کلاستر
۲. تعادل بارگذاری شبکه Network Load Balancing

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. سرورهای در کلاسترینگ نود نیز نامیده میشوند و همگی نود ها از یک سری برنامه و داده ی خاص و دیگر امکانات شبکه استفاده میکنند. لذا هر کدام از نودها میتوانند درخواست های کلاینت را انجام دهند. تنظیم کردن نود که بتواند فعال Active یا غیرفعال Passive (برای مواقع ضروری فعال شود) دست خود شماست.

برای مثال یک سرور کلاستر ساده میتواند شامل دو نود باشد که هر دو باید ویندوز سرور۲۰۰۳ و Microsoft SQL را در حالت اجرا داشته باشند و هر دو به NAS شبکه دستری داشته باشند که اطلاعات مربوط به بانک داده را درون خودش ذخیره کرده است. یکی از نود ها باید Active و دیگری باید Passive باشد. در حالت معمول نود فعال به صورت عادی به کار خود ادامه میدهد به درخواست های کلاینت ها پاسخ میدهد و اطلاعات را انتقال و یا دریافت میکند اما اگر نود فعال به هر دلیلی از سرویس خارج شد نود غیرفعال به سرعت از سرویس خارج شدن نود فعال را تشخیص میدهد و خود فعال میشود و به درخواست های کلاینت ها رسیدگی میکند. البته این روش یک مشکل عمده دارد که در این روش Ative/Passive همیشه یکی از سرورها بی کار میباشد و تا زمانی که سرور اول از سرویس خارج نشود کاری انجام نمیدهد اما بسته به نوع برنامه میتوان سرورکلاستر ها را طوری طراحی کرد که بتوانند کارهای مربوط به برنامه و وظایف را بین یکدیگر تقسیم کنند و بازدهی سرویس را بالا ببرند.



جدای از نودهای درون کلاستر که در حال اجرا هستند یک سرور کلاستر پروتکل و نام اختصاصی خود را دارد بنابراین اگر سیستمی با خطا مواجه شود تغییر در عملکرد سرور برای کلاینتی که درخواست خود را صادرکرده است احساس نمیشود و کلاینت درخواست خود را به همان مقصد ارسال میکند. در چنین موقعی سرور غیرفعال درخواست را گرفته و رسیدگی میکند بنابراین کاهشی در کارایی و اجرای دستورالعمل ها وجود ندارد.

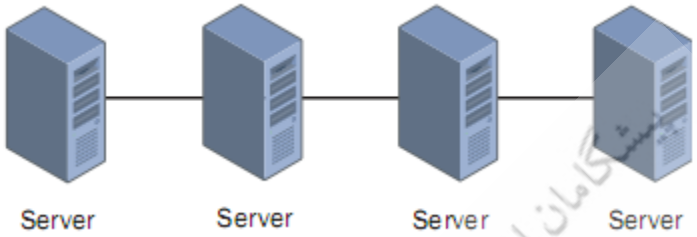
نکته ای که باید در نظر داشت این است که نوع ویندوز هایی که کامپیوترهای یک کلاستر اجرا میکنند نیز باید یکی باشد.

Balancing Network Load

Balancing Network Load

نوعی از کلاسترینگ میباشد که هم قابلیت دستیابی بالاتر و راحت تر و هم ظرفیت بیشتری دارد. NLB برای برنامه هایی طراحی شده است که وابستگی داده ای کمتری دارند و این داده ها کمتر تغییر میکنند که گاهاً حتی ممکن است فقط خواندنی نیز باشند و زمانی زیادی را در حافظه باقی نمانند. به این برنامه ها Stateless گفته میشود و معمولاً دارای پروتکل FTP و سرورهای VPN هستند. درخواست هر کلاینت به برنامه های Stateless یک فعل و انفعال جدا محسوب میشود پس میتوان درخواست ها را در میان تمامی سرورها توزیع کرد تا بتوان لد مورد پردازش برنامه ها را دارای تعادل کرد.

به جای داشتن یک منبع اطلاعات؛ در یک NLB کلاستر، تمامی سرورهای برای خود منبع اطلاعاتی یکسانی دارند و همگی نیز از نوع سرورهای فعال هستند. نرم افزار کلاسترینگ درخواست کلاینت را بین نودها توزیع میکند. هر کدام از نود ها که درخواست های خود را مستقلانه پردازش میکند از اطلاعات محلی خود استفاده میکند و هنگامی که یکی از سرورها خراب شود بقیه ی سرورها مقداری از درخواست های ارسال شده به سرور خراب را برمیدارند و اجرا میکنند.



هنگامی که یک سرور به فرض یک رکوردی را دریافت میکند و آن را در دیتابیس ذخیره میکند بقیه ی سرورها اجازه ی دسترسی به آن رکورد را ندارند تا زمانی که بقیه ی سرورها نیز آن رکورد را در دیتابیس خود ذخیره کنند.هنگامی که یک سرور به فرض یک رکوردی را دریافت میکند و آن را در دیتابیس ذخیره میکند بقیه ی سرورها اجازه ی دسترسی به آن رکورد را ندارند تا زمانی که بقیه ی سرورها نیز آن رکورد را در دیتابیس خود ذخیره کنند.

NLB علاوه بر دسترسی بالاتر و راحتی بیشتر قابلیت بزرگتر شدن مقیاس را نیز دارد. هنگامی که تعداد درخواستها بیشتر میشود تنها کاری که میشود این است تعداد سرورهای کلاستر را بیشتر کرد که در این صورت هر سرور تعداد درخواست کمتری را پردازش میکند. تمامی نسخه های ویندوز ۲۰۰۳ برای هر کلاستر ۳۲ سرور را پشتیبانی میکنند.

پیاده سازی و کلاستر کردن

اولین نکته ای که باید قبل از تصمیم به کلاستر کردن یک سری سرور مد نظر داشته باشید این است که هدف شما از کلاستر کردن چیست و چه انتظاری از کلاسترینگ خود دارید یا به عبارت دیگر چقدر به دسترسی بالاتر، راحتی بیشتر و یا مقیاس بزرگتر نیاز دارید. برای برخی از سازمان ها که

دسترسی بالاتر و بیشتری را میطلبند کلاسترینگ با دسترسی بیشتر به این معنی است که از کار افتادن سرورها و برنامه ی آنها به هیچ وجه قابل قبول نیست و کلاسترینگ باید راه حلی برای سه مشکل عمده ی زیر پیدا کند:

۱. **خطاهای نرم افزاری:** بسیاری از خطاها ممکن است باعث این شود که یک برنامه به درستی کار نکند. خود برنامه با درست کار نکردن میتواند یک دلیل باشد، بخشی از اجزای سیستم میتواند باعث این کار شود یا اینکه سیستم عامل یک سری مشکلات را داشته باشد که باعث شود تمامی برنامه ها با مشکل اجرا شوند. مشکلات نرم افزاری میتواند با آپگرید کردن سخت افزاری نیز رخ دهد، ممکن است با برنامه های جدید که روی سیستم نصب میشود همخوانی نداشته باشد و البته یا با وجود ویروس یا هر کد مخرب دیگری درون سیستم. اما تا زمانی که مدیر اقدامات و پیشگیری های لازم را انجام دهد (برای مثال نصب نکردن آپدیت های نرم افزاری در تمامی سرورهای یک کلاستر) یک کلاستر میتواند برنامه را در دسترس کلاینت ها قرار دهد هرچند که ممکن است یک یا چند عدد از سرورهای کلاینت با برنامه مشکل داشته باشند.
۲. **مشکلات سخت افزاری:** هارد درایو، فن های خنک کننده، منابع تغذیه و بقیه ی اجزای سخت افزاری میتواند باعث مشکل شود که در این حالت کلاسترینگ برنامه ی کمکی را اجرا میکند در صورتی که ممکن است خطایی سخت افزاری رخ داده باشد. علاوه بر این کلاسترینگ میتواند این امکان را به مدیران میدهد که بتوانند کارهای مربوط به نگهداری سخت افزاری را انجام دهند بدون این که مجبور به از سرویس خارج کردن یک برنامه حیاتی باشند.
۳. **مشکلات مکانی:** در یک کلاستر که از نظر جغرافیایی دارای سرورهای پراکنده میباشد، سرورها ممکن است در ساختمان های مختلف یا شهرهای مختلف باشند، صرف نظر از اینکه با این کار میتوان نرم افزارهای حیاتی را در نقاط مختلف برای کاربران مختلف قابل دسترس

ساخت مزیت دیگر این روش این است که اگر هر حادثه ای همانند آتش سوزی یکی از مکان های کلاسترینگ را از بین ببرد باز هم کلاسترینگ برنامه را در دسترس قرار میدهد.

میزان دستیابی بالا High Availability

میزان دستیابی که شما نیاز دارید به گوناگونی فاکتورهای شما بستگی دارد که شامل نوع برنامه هایی که اجرا میکنید، اندازه، مکان و تقسیم مکانهای کاربران و نقش برنامه ها در سازمان شما میباشد. در برخی از موارد داشتن یک برنامه ای که همیشه در دسترس کاربران باشد (همیشه در حال اجرا باشد) برای راحتی بیشتر خواهد بود اما در حالتی دیگر این امر ضروری میباشد. مقدار قابل دستیابی بودن یک برنامه در یک سازمانی که به آن برنامه نیاز دارد میتواند به روش های مختلف در تنظیم و پیکربندی کلاسترینگ آن نقش داشته باشد که از جمله ای انها میتوان به نوع کلاستری که استفاده میشود، تعداد سرورها در کلاستر، توزیع برنامه ها روی سرورهای کلاستر و مکان سرورهای کلاستر اشاره کرد.

در دسترس بودن معمولاً به صورت درصدی از زمان بیان میشود که یک برنامه میتواند در حالت اجرا باقی بماند برای مثال Availability قابل دسترس بودن ۹۹٪ به این معنی است یک برنامه برای مدت ۸۷.۶ ساعت از کل یک سال را نمیتواند در حال اجرا باشد یعنی برنامه ای که ۹۹٪ قابلیت در دسترسی بودن را دارد نمیتواند بیشتر از ۸۷.۶ ساعت از اجرا خارج شود. برای به دست آوردن میزان دسترسی نمیتوان فقط به پیاده سازی کلاسترینگ توجه داشت بلکه برای اینکار نیاز به نصب سخت افزار جانبی دیگر هم دارید که بتواند میزان کارایی سخت افزار و نرم افزار را ارزیابی و آنها را تست کند و همچنین سیاستهایی را برای کل ساختمان IT خود پیاده سازی کنید. به مجرد اینکه تقاضای دسترسی به برنامه بالاتر میرود میزان زمان، پول و تلاشی که برای دستیابی به آن میزان دسترسی مورد نیاز است نیز بیشتر میشود. ممکن است فکر کنید که برای مثال رسیدن از ۹۵٪ راحتی Reliability به ۹۹٪ کار راحتی باشد اما در واقع قیمت نهایی بسیار گران تمام خواهد شد

Scaling Clusters افزایش مقیاس کلاستر

هر دو نوع کلاسترینگ قابلیت پیاده سازی در مقیاس های بزرگتر را دارند بدین معنی که شما میتوانید کارایی کلاستر خود را مطابق نیاز خود و میزان رشد سازمان خود بالا ببرید.

Sacling Up : افزایش کارایی سرورها با ارتقا دادن سخت افزار آنها، اضافه کردن RAM، یا کش L2، قرار دادن CPUهای سریعتر از راه هایی هستند که میتوان به افزایش کارایی کلاستر کمک کند. بهبود کارایی در این روش به نوع پیاده سازی کلاستر شما نیز وابسته میباشد. و شما باید این میزان افزایش توانایی را برای تمامی سرورها اعمال کنید چرا که ممکن است کامپیوتر فعال دارای قدرت بیشتری باشد و زمانی که از کار میافتد کامپیوترهای غیرفعال نتوانند میزان کار درخواست شده را انجام دهند بنابراین باید تمامی سرورهای کلاستر در یک سطح از توانایی قرار داشته باشند.

Scaling Out : اضافه کردن سرور به کلاستر. زمانی که شما یک سری درخواست را بین سرورها توزیع میکنید با افزایش سرورها این میزان پردازش با تقسیم شدن کمتر میشود، افزایش سرور را میتواند در هر دو نوع سرورکلاستر و NLB انجام داد.

اضافه کردن یک سرور جدید به NLB کار راحتی میباشد به این علت که هر سیستم در این کلاستر دارای منبع اطلاعات مستقل است لذا اضافه کردن یک سرور جدید کاری راحت میباشد، به طوری که با کپی کردن منبع اطلاعات و برنامه درون سرور جدید میتوان آن را به کلاستر اضافه نمود

اضافه کردن یک سرور جدید به کلاستر سرور کمی پیچیده تر از NLB میباشد، چرا که در این نوع کلاستر تمامی سرورها از یک منبع اطلاعاتی مشترک استفاده میکنند، با توجه به تنظیمات سخت افزاری شما افزایش مقیاس میتواند بسیار گران و یا شاید هم غیر ممکن باشد.

محدودیت دیگری هم برای افزایش مقیاس کلاسترها وجود دارد و آن نوع سیستم عاملی است که در سرورها مورد استفاده قرار میگیرد، زمانی که اقدام به افزایش یک سرور جدید میکنید باید ببینید سیستم عامل شما از چه ماکزیمم سروری پشتیبانی میکند. لیست زیر نوع سیستم عامل و تعداد ماکزیمم سرورهایی که پشتیبانی مکنند را نمایش میدهد:

Server Clusters	NLB	سیستم عامل
پشتیبانی نمیکند	32	Windows Server 2003, Standard Edition
8	32	Windows Server 2003, Enterprise Edition
8	32	Windows Server 2003, Datacenter Edition
2	32	Windows 2000 Advanced Server
4	32	Windows 2000 Datacenter Server

همچنین هنگام ارتقای سخت افزاری سرورها نیز محدودیت هایی از طرف سیستم عامل وجود دارد که به شرح زیر است:

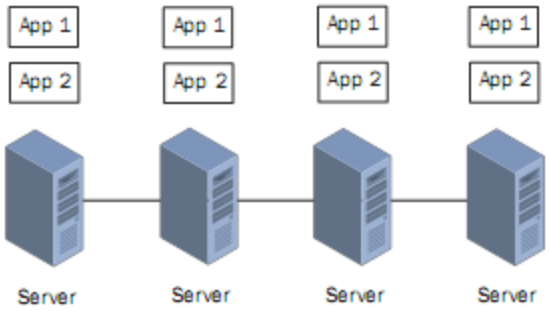
بیشترین مقدار RAM	بیشترین تعداد پردازنده	سیستم عامل
4 GB	2	Windows Server 2003, Standard Edition
32 GB	8	Windows Server 2003, Enterprise Edition
64 GB	32	Windows Server 2003, Datacenter Edition
8 GB	8	Windows 2000 Advanced Server
64 GB	32	Windows 2000 Datacenter Server

چه تعداد کلاستر ؟

اگر شما قصد اجرای بیش از یک برنامه را با قابلیت دسترسی بالا داشته باشید باید تصمیم بگیرید که چه تعداد کلاستر میخواهید داشته باشید. سرورکلاسترها میتوانند چندین برنامه را به صورت همزمان اجرا نمایند بنابراین شما میتوانید تعدادی برنامه را روی یک سرور کلاستر اجرا نمایید یا اینکه سرور کلاسترهای جدا ایجاد نمایید تا هر کدام یک برنامه را اجرا نمایند.

کلاستر سرور جدا برای برنامه ها

برای مثال شما دو برنامه Statefull دارید و قصد اجرای این برنامه ها را با استفاده از سرورکلاستر دارید، ساده ترین روش این است که یک سرورکلاستر ایجاد کنید و هر دو برنامه را روی سرورهای آن اجرا نمایید. اگر یکی از کامپیوترها با خطا مواجه شده و از سرویس خارج شود باقی سرورها باید توانایی اجرای برنامه های آن را داشته باشند.



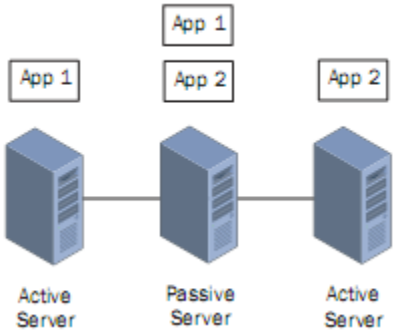
ترکیبی از دو حالت قبل

روش دیگر ایجاد کلاستر سرور جدا براي هر برنامه است که سرورهاي هر کلاستر به صورت جداگانه کار میکنند و خطایی در هر کلاستر باعث تاثیر گذاري روي برنامه ي همان کلاستر میشود، شبه علاوه سرورهاي باقی مانده تنها یک برنامه را سرویس دهی مینمایند. ایجاد سرورهاي جدا از قابلیت دسترسی بالاتري برخوردار است اما پیاده سازي آن گرانتر تمام میشود چرا که نیاز به تعداد سرورهاي بیشتري است.

ترکیبی از دو حالت قبل

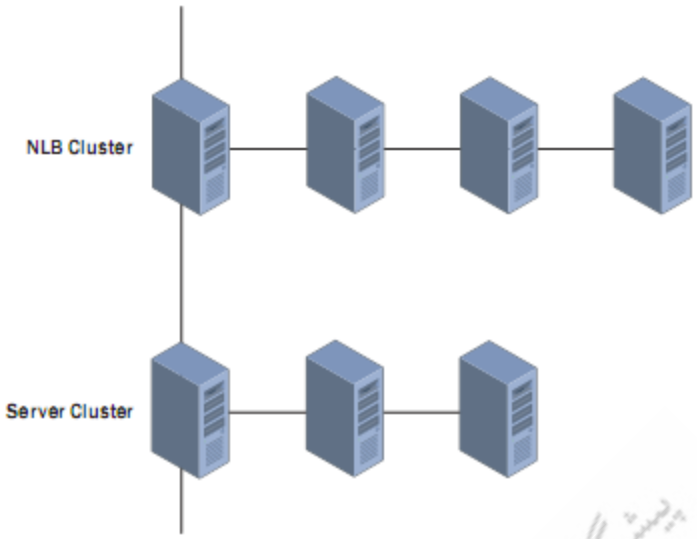
علاوه بر این شما میتوانید ترکیبی از این دو حالت را نیز داشته باشید طوری که یک کلاستر سرور ایجاد کنید که در ان دو سرور فعال داشته باشید و هر کدام از برنامه ها رو روی یک سرور فعال نصب نمایید. و یک سرور غیرفعال داشته باشید که به عنوان پشتیبان هر دو برنامه را روی آن نصب نمایید. در صورتی که یکی از سرورهای فعال از سرویس خارج شود سرور غیرفعال وظیفه ی آن را به عهده میگیرد در صورتی که هر دو سرور فعال از سرویس خارج شوند سرور غیرفعال به جای هر دو سرور

دیگر سرویس دهی مینماید. بستگی به شما دارد که توانایی و ظرفیت سرور غیرفعال در حد اجرای یک برنامه باشد یا هر دو برنامه.



ادغام دو تکنولوژی کلاسترینگ سرور کلاستر و NLB با هم

تصمیم برای اینکه از چه نوع کلاسترینگی (NLB یا Server Clusters) برای کلاستر کردن خود استفاده میکنید به نوع برنامه ی شما بستگی دارد. اما در برخی مواقع برای بازدهی بهتر میتوان کلاسترها را با هم دیگر ترکیب نمود و برنامه ها را با کیفیت بهتری از لحاظ Availablity اجرا نمود. بهترین مثال برای این گونه سرورها میتواند سیستم خرید الکترونیکی باشد که وب سایت به کاربران اجازه میدهد به صورت آنلاین خرید نمایند. در این نوع طراحی وب سرورها (که از نوع برنامه های Stateless هستند) اجرای خود سایت را به عهده میگیرند و سرورهای دیتابیس (که از نوع برنامه های Statefull هستند) وظیفه ی ذخیره ی اطلاعات مشتریان، محصولات و سفارشات خرید آنها را ذخیره میکنند و شما میتوانید از کلاستر NLB برای میزبانی وب سرورها استفاده کنید و از Server Cluster ها برای سرورهای بانک اطلاعاتی استفاده نمایید.



پیشگامان امن داده های پایدار (پادکو)
PADCO - Datacenter

استفاده از Balancing Network Load

با توجه به اینکه هر دو نوع کلاسترینگ کردن توسط ویندوز ۲۰۰۳ پیشتیبانی میشود اما استفاده از NLB آسان تر خواهد بود به این دلیل که دارای نصب، پیکربندی راحت تری است به علاوه ی اینکه شما میتوانید از سخت افزار موجود نیز استفاده کنید و همچنین نیاز به نرم افزار اضافه ای بر روی کامپیوترهای خود ندارید. میتوانید با استفاده از برنامه ی NLB Manager در ویندوز ۲۰۰۳ به روی سرورهای کلاستر خود مدیریت داشته باشید.

مفهوم NLB

یک NLB کلاستر شامل ۳۲ کامپیوتر میباشد که همگی از یک سری اطلاعات کپی شده روی منبع اطلاعات خود استفاده میکنند که کلاینت ها این سرورها را به عنوان هاست شناخته و به این سرورها مراجعه میکنند. NLB طوری کار میکند که روی هر یک از هاست ها یک Virtual Network

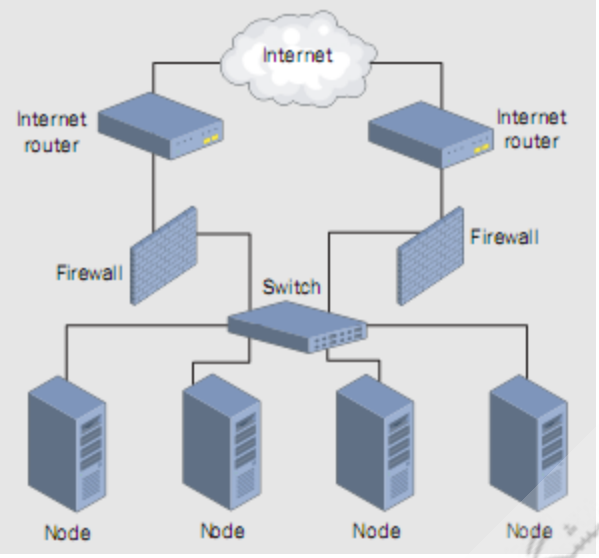
Adapter نصب میکند که کلاستر را به صورت یک موجودیت نمایان میکند. Virtual Adapter آدرس آی پی و آدرس شبکه یا MAC منحصر به فرد خود را دارد که مستقل از آدرس فیزیکی است که به کامپیوترها در شبکه داده شده است و کلاینت ها درخواست خود را به آدرس کلاستر میفرستند به جای اینکه به آدرس یک سرور خاص بفرستند.

هنگامی که یک درخواست از طرف یک کلاینت به آدرس آی پی کلاستر فرستاده میشود و از طرف کلاستر دریافت میشود این درخواست به تمامی سرورهای درون کلاستر ارسال میشود. روی هر کدام از سرورهای NLB کلاستر، NLB به عنوان یک ف**ی**ل**ت**ر بین رابط کلاستر پشته های پروتکل های TCP/IP نودها عمل میکند. این ف**ی**ل**ت**ر این امکان را میدهد که NLB تشخیص دهد کدام درخواست را کدام سرور انجام دهد. در این حالت هیچ ارتباطی بین سرورها وجود ندارد و هر یک به طور مستقل از هم کار میکنند و درخواست ها را پردازش میکنند.

قبل از پیاده سازی یک NLB باید ساختار شبکه ی خود را برای سرورهای کلاستر مشخص کنید. قابلیت دسترسی بالایی که این کلاستر در اختیار قرار میدهد چندان راضی کننده نیست برای اینکه ممکن است کاربران شما در هنگام از سرویس خارج شدن یک روتر، سویچ یا قطع شدن اینترنت دسترسی خود را به برنامه از دست بدهند، از آنجایی که بسیاری از سرویس های وب و اینترنت توسط این کلاستر پشتیبانی میشود باید امنیت کاربران اینترنتی در آن برقرار باشد و شما باید به این امر نیز رسیدگی کنید.

طراحی NLB

در یک شبکه با ترافیک بالا که قابلیت دسترسی بالا نیز دارد، این شبکه با چندین ارتباط به اینترنت وصل شده است که این اتصالات میتواند از ISP های مختلف باشد. در هر کدام از روترها یک Firewall شبکه را از خطرات اینترنتی حفظ میکند و Firewall دیگر نیز امنیت داخلی شبکه را به عهده دارد.



نوع عملکرد NLB:

سرورهایی که میخواهید برای کلاستر کردن به روش NLB استفاده کنید نیاز به داشتن سخت افزاری قدرتمند ندارند. چیزی به نام مخزن اطلاعات یا داده نیز در این روش وجود ندارد که بین سرورهای کلاستر Share شود. برای مثال شما نیازی به این ندارید که یک مخزن ذخیره ی اطلاعات در شبکه داشته باشید. هرچند که NLB خودش نوعی محدودیت را روی رابط سرورهای شبکه درحالت استاندارد اعمال میکند، در برخی مواقع شما میتوانید از روش دومی نیز برای برقراری ارتباط سرور با شبکه بهر ببرید.

ویندوز سرور ۲۰۰۳ برای NLB دو نوع عملکرد را دارا میباشد : روش Unicast و روش Multicast. NLB آدرس نقاب آدرس فیزیکی هر سرور را با آدرس نقاب آدرس های مجازی سرورها تعویض میکند و به هیچ وجه از آدرس نقاب آدرس فیزیکی نود ها استفاده نمیکند و در واقع ارتباط واقعی کامپیوترهای کلاستر به یک ارتباط مجازی تبدیل میکند. Address Resolution Protocol یا ARP عمل رفع کردن آدرس آی پی سرورها را به یک آدرس نقاب واحد انجام میدهد.

ARP نوعی پروتکل TCP/IP میباشد که آدرس های IP را به آدرس های نقاب و یا سخت

افزاری تبدیل میکند. برای انتقال یک درخواست به یک آدرس IP خاص کامپیوتر باید آدرس نقابی را که وابسته به آدرس IP میباشد را پیدا کند. ARP پیغامی را شامل یک آدرس IP به کل شبکه ارسال میکند و کامپیوتری که آدرس IP آن با آدرس داخل پیغام یکی باشد باید آدرس MAC خود را با استفاده از یک پیغام بازگرداند. در حالت Unicast در کلاستر NLB هر سرور به درخواست ARP که فرستاده شده اگرچه پیغام شامل آدرس آیپی همان کامپیوتر باشد یا آدرس آیپی خود کلاستر به درخواست پاسخ خواهد داد و آدرس نقاب کلاستر را باز میگرداند.
به دلیل این که آداپتورهای شبکه ی تمامی سرورهای شبکه ی کلاستر از یک آدرس نقاب میباشد لذا سرورهای درون کلاستر به صورت نورمال نمیتوانند با یکدیگر ارتباط برقرار کند. البته سرورها میتوانند با دیگر کامپیوترهای همان Subnet یا هر Subnet دیگری ارتباط برقرار کنند به شرط اینکه دیتاگرام آدرس IP آنها شامل آدرس نقاب نباشد.
در بیشتر مواقع این باعث بروز مشکلی نمیشود و معمولاً سرورها و هاست ها نیازی به ارتباط با یکدیگر در شرایط معمولی پیدا نمیکنند. با این حال اگر از نظر شما سرورهای درون کلاستر شما باید توانایی برقراری ارتباط با یکدیگر را داشته باشند این کار به دو روش امکان پذیر است:

۱. تنظیم سرورهای کلاستر به استفاده از از روش Multicast. در روش چندپخشی؛ NLB یک آدرس نقاب کلاستر به آداپتر رابط شبکه اختصاص میدهد اما آدرس نقاب اصلی آداپتر رابط شبکه را نیز نگه داری میکند. برای درست کار کردن این روش روتر باید از آدرسهای نقاب چندپخشی پشتیبانی کند.
۲. نصب یک آداپتور رابط شبکه روی هر یک از سرورها، یکی از آداپتورها به همراه آدرس اصلی نقاب آن که با آدرس نقاب کلاستر جایگزین شده است آداپتور کلاستر میشود.

به طور خلاصه یک سرور در کلاستر NLB هم میتواند یک آداپتور رابط شبکه داشته باشد و هم بیش از یک آداپتور. همچنین هم میتواند به روش تک پخشی و چندپخشی اجرا شود. با ترکیب این تنظیمات ۴ حالت ممکن برای پیکربرندی NLB بوجود می آید که هر یک ویژگی ها و مضرات خود را دارد.

NLB Configuration	Advantages	Disadvantages
Single network interface adapter in unicast mode	■ Requires no special hardware ■ No router incompatibility problems	■ Ordinary communications with other servers in the cluster are not possible ■ Network performance might degrade when one network interface adapter is handling both ordinary traffic and cluster traffic
Single network interface adapter in multicast mode	■ Requires no special hardware ■ Permits ordinary communications among cluster servers	■ Some routers cannot support multicast MAC addresses ■ Network performance might degrade when one network interface adapter is handling both ordinary traffic and cluster traffic
Multiple network interface adapters in unicast mode	■ No router incompatibility problems ■ Permits ordinary communications among cluster servers ■ Network performance enhanced, because cluster traffic and ordinary network traffic use different network interface adapters	■ Requires installation of second network interface adapter
Multiple network interface adapters in multicast mode	■ Permits ordinary communications among cluster servers ■ Network performance enhanced, because cluster traffic and ordinary network traffic use different network interface adapters	■ Requires installation of second network interface adapter ■ Some routers cannot support multicast MAC addresses

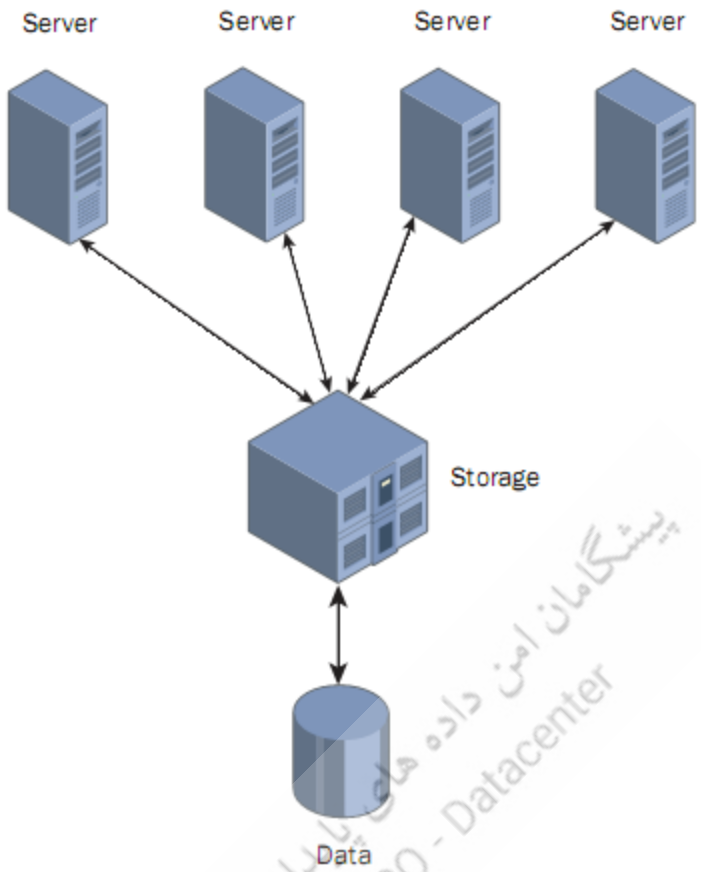
PADCO - Datacenter

بیشترین پیکربندی که در نصب NLB استفاده میشود نصب دو آداپتور رابط شبکه در هر سرور و اجرای آنها به روش تک پخشی Unicast میباشد. این نوع پیکربندی این امکان را میدهد که سرورها هم بتوانند به عنوان یک شرکت کننده ی معمولی در شبکه حضور داشته باشند و هم بتواند وظایف خود را به عنوان یک سرور کلاستر انجام دهند. مشکلی هم با به دست گیری آدرسهای نقاب چندپخشی ندارد و هیچ مشکلی از لحاظ ترافیک نیز وجود نخواهد داشت.

Server Cluster

طراحی Cluster Server

سرورکلاسترها از کلاسر NLB پیچیده تر هستند هم در اداره ی برنامه و هم در اداره ی منبع اطلاعات. همانطور که گفته شد هنگامی که قصد پیاده سازی یک سرورکلاستر را دارید باید به میزان قابل دسترس بودن برنامه ی خود توجه داشته باشید اما باید این نکته را هم مد نظر داشته باشید که سرورکلاستر با افزایش ظرفیت گران تر هم خواهد شد.ضمناً سرورکلاستر برای برنامه هایی به کار میرود که بیشتر در حافظه قرار دارند همانند سرورهای ایمیل یا بانک اطلاعاتی، چراکه اطلاعاتی که در حافظه ی این سرورها قرار دارد بارها و بارها عوض میشود. این امر در صورتی که سرورها از منبع اطلاعاتی مستقلی استفاده میکردند اصلاً به صرفه نبود چرا که اگر چنین بود با هر درخواستی از طرف کلاینت که نتیجتاً تغییری را در اطلاعات یکی از سرورها ایجاد مینمود باید تمامی سرورهای دیگر به سرعت اطلاعات خود را مطابق با آن تغییر میدادند که کار و زمان زیادی از سرورها گرفته میشد. به طور کلی میتوان گفت که سرورکلاستر از خاصیت Share کردن یک منبع اطلاعاتی واحد برای تمامی سرورها استفاده میکند و کلاستر اطلاعات مربوط به بانک داده یا ایمیل را درون آن ذخیره میکند و این منبع اطلاعاتی به تمامی سرورهای درون کلاستر متصل است، لذا تمامی برنامه های کلاینتها صرف نظر از اینکه به کدام سرور وصل شده اند اطلاعات خود را یک از منبع اطلاعاتی میگیرند.



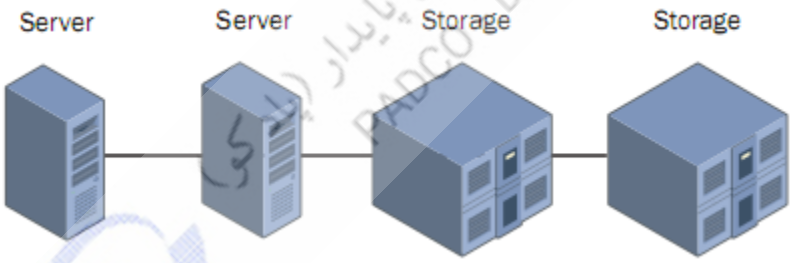
اگر قصد ایجاد یک کلاستر پراکنده را دارید بر خلاف NLBکلاستر که تمامی سرورها با استفاده از یک تکنولوژی خارجی همانند Round Robin DNS با یکدیگر در ارتباط هستند در سرورکلاستر تمامی سرورها حتماً باید به منبع اطلاعاتی متصل باشند حتی اگر در شهرهای مختلف قرار دارند.

تنظیمات سخت افزاری یک سرورکلاستر

تمامی کامپیوترهایی که ویندوز۲۰۰۳ روی آنها نصب است و شما میخواهید از آنها به عنوان سرورهای کلاستر خود استفاده کنید همگی باید از یک معماری پردازنده استفاده کنند به این معنی که نمیتوان سیستم های ۳۲ بیتی و ۶۴ بیتی را با هم ترکیب کنید. هر سرور درون کلاستر باید حداقل یک اتصال به شبکه ی استاندارد داشته باشد که امکان دسترسی به باقی سرورهای کلاستر و کلاینتها را بدهد. برای داشتن قابلیت دسترسی بالاتر Availability High، داشتن دو اتصال به شبکه بهتر از

یک اتصال است که یکی به شبکه ی کلاینتها متصل است و دیگری به شبکه ای که بین سرورهای دیگر ارتباط ایجاد میکند متصل است. علاوه بر اتصال استاندارد به شبکه هر سرور باید یک اتصال جدا نیز به منبع اطلاعاتی داشته باشد. ویندوز سرور۲۰۰۳ سه نوع ارتباط را تعریف میکند: SCSI و دو نوع کانال فیبری که در ادامه ی بحث توضیح داده میشوند.

SCSI : نوعی Bus میباشد که برای ارتباط دستگاه های ذخیره و دیگر دستگاه های جانبی به کامپیوتر شخصی به کار میرود. پیاده سازی SCSI به نوعی در شکلبندی آداپتور هاست ها به کار میرود و تعدادی از دستگاه های ورودی و خروجی که شما به کارت ارتباط میدهید همگی از یک کابل SCSI مناسب استفاده میکند. در پیکربندی یک باس SCSI، شما از چندین آداپتور شبکه استفاده میکنید که برای هر سرور در کلاستر یکی مورد نیاز است و آداپتورها و منابع اطلاعاتی را به یک Bus متصل میکنید.



به دلیل محدودیت معماری Bus، ویندوز سرور۲۰۰۳ تنها از دو SCSI نود در کلاستر پشتیبانی میکند و تنها نسخه ی ۳۲ بیتی ویندوز۲۰۰۳ از این قابلیت پشتیبانی میکند به علاوه که شما از SCSI برای کلاسترهای پراکنده استفاده کنید چرا که ماکزیمم متراژی که SCSI پشتیبانی میکند ۲۵ متر است.

استفاده از Fiber Channel، کانال فیبر

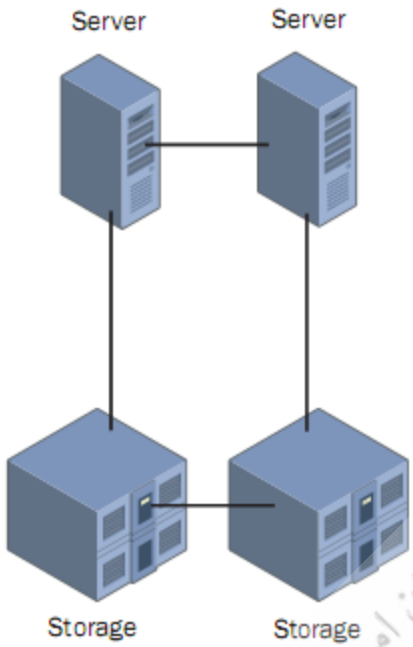
کانال فیبر تکنولوژی ای با سرعت بالا میباشد که معمولاً برای ارتباط بین کامپیوترها و دستگاه های

ذخیره سازی (Storage Device) مورد استفاده قرار میگیرد. برخلاف SCSI که از تکنولوژی موازی استفاده میکرد در این تکنولوژی اطلاعات به صورت سری رد و بدل میشوند لذا این امر امکان انتقال داده به مصافت های دورتر را فراهم میسازد. کانال فیبر میتواند اطلاعات را با سرعت ۱۰۰ مگابیت در ثانیه در حالت Full Duplex انتقال دهد؛ این بدین معنی است که اطلاعات میتوانند با سرعت بالایی به صورت همزمان و در هر دوجهت رفت و برگشت جابه جا شوند.
معمول ترین روش برای پیاده سازی یک کانال فیبر این است که آن را به صورت یک هاست آداپتور در کلاستر سرور نصب کرد و سپس از آنها برای ارتباط کامپیوترها با دستگاه های ذخیره استفاده کنند. ویندوز سرور۲۰۰۳ از دو نوع کانال فیبر برای اتصال به مخازن اطلاعاتی پشتیبانی میکند.

۱. Fiber Channel Arbitrated Loop
۲. Fiber Channel Switched Fabric

Fiber Channel Arbitrated Loop

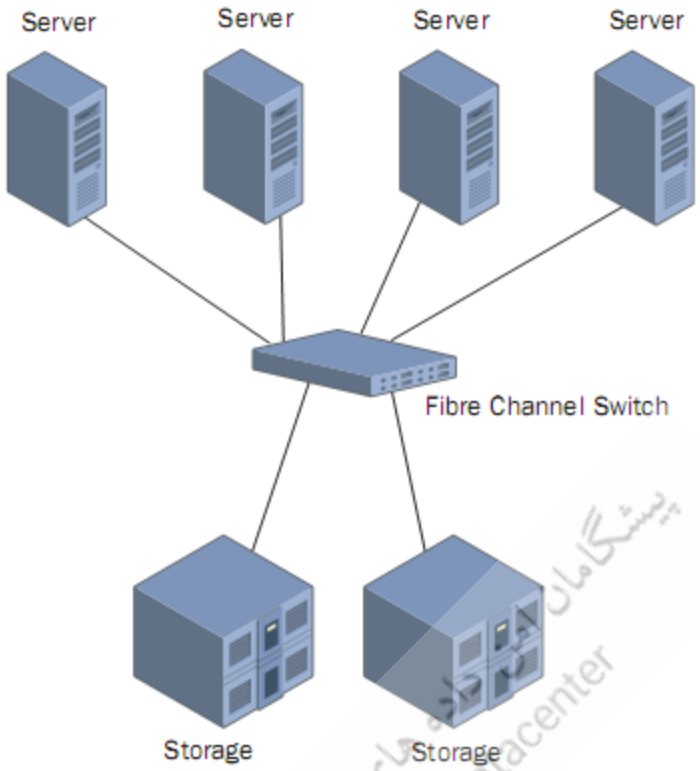
در مفهوم سرورکلاستر ویندوز سرور۲۰۰۳ یک FCAL یک تکنولوژی حلقه ای است که سرورهای کلاستر را به تعدادی از مخازن اطلاعاتی (Storage Device) متصل میکند.تعداد دستگاه ها در این حالت محدود به ۱۲۶ میباشد اما ویندوز سرور۲۰۰۳ تعداد سرورها را در یک FCAL به ۲ محدود کرده است.



یک FCAL نوعی رسانه ی تسهیم اطلاعات در شبکه میباشد. اطلاعاتی که از یک دستگاه یا مخزن اطلاعات به سروری منتقل میشود ممکن است از میان دستگاه های دیگر عبور کن که باعث کاهش پهنای باند یک دستگاه دیگر روی شبکه میشود. در مقایسه با FCSF این تکنولوژی کلاستر کردن سخت افزاری، زیاد گران قیمت نیست و این امکان را به Admin میدهد که ظرفیت مخازن اطلاعاتی خود را به راحتی افزایش دهد البته نمیتواند نود دیگری را به کلاستر اضافه کند.

Fabric Fiber Channel Switched

تنها روشی که به شر کردن مخازن اطلاعاتی میپردازد و ویندوز۲۰۰۳ نیز تعداد سرورهای بیشتر از ۲ را برای کلاستر کردن آن پشتیبانی میکند FCSW میباشد. این روش بیشتر شبیه یک سوئیچ شبکه عمل میکند به این صورت که همه دستگاه ها به یک سوئیچ متصل شده اند. سوئیچ کردن این امکان را به دیگر دستگاه های کلاستر میدهد که بتوانند به طور مستقیم با دیگر دستگاه ها ارتباط برقرار کنند، در این روش هیچ رسانه ی Share کردن برای شبکه وجود ندارد و تمام پهنای باند برای ارتباطات داخل شبکه میباشد.



یک شبکه ی FC-SW که دسترسی کاملی را به سرورها برای دستیابی به منبع اصلی اطلاعات میدهد نوعی از SAN میباشد. ایجاد یک SAN برای سرویس دهی به سرورهای خود در کلاستر بیشتر انعطاف پذیری و افزاش مقایس را در پی خواهد داشت. میتوانید با استفاده از نصب کردن سرورها و متصل کردن آنها به کلاستر، نودهای بیشتری را اضافه کنید یا ظرفیت مخزن اطلاعاتی خود را با نصب درایوهای بیشتر افزایش دهید همچنین با استفاده از SAN میتوانید یک شبکه ی پراکنده داشته باشید حتی در شهرهای مختلف.

SAN به معماری ویژه ای گفته میشود که در آن مخازن اطلاعاتی به گونه ای به سرورها متصل میشوند که سیستم عامل این دستگاهها را به صورت Attached Local میبینند.

نصب برنامه روی سرور

برنامه های Stateful که هاست های سرورکلاستر معمولاً اجرا مینمایند از برنامه ای Stateless که به روی NLB ها اجرا میشود ظرفیت و قابلیت بیشتری دارند این به این معنی است که انعطاف بیشتری در نصب و پیاده سازی برنامه روی کلاستر وجود دارد. ویندوز سرور۲۰۰۳ دو نوع برنامه را برای سرورکلاسترها پیشتیبانی میکند.

برنامه ی Single-Instance : برنامه هایی که در یک لحظه فقط میتواند روی یک سرور با استفاده از تنظیماتی که برای برنامه تعیین شده است اجرا شود. نمونه ی ساده ای که از این برنامه ها میتواند مثال زد سرویس DHCP میباشد. شما تنها بر روی یک سرور میتوانید یک DHCP را با تنظیمات Scope آن اجرا نمایید مگر اینکه ریسک این که احتمالاً آدرس IP مشابه را روی شبکه ی خود قبول کنید! برای اجرای این نوع برنامه ها، برنامه فقط باید روی یک نود اجرا شود و باقی نودها در حالت Stand by قرار دارند. اگر نود فعال از سرویس خارج شود یکی از نودهای دیگر برنامه را اجرا خواهد کرد.

برنامه های Multiple-Instance : برنامه هایی با کدهای مشابه یا کپی شده که میتوانند همانند سرورهای NLB روی چندین سرور اجرا شوند. البته برنامه هایی که بتوان کدهای آن را به بخش های مختلف تقسیم کرد نیز از این قبیل برنامه ها محسوب میشوند تا بتوانند سرویسهای تکمیلی را به همراه نودهای دیگر کلاستر انجام دهند. با برخی از بانکهای اطلاعاتی میتوان بخش های مختلفی را ایجاد کرد تا به یک درخواست خاص سرویس دهند.

به کار بردن برنامه های Single Instance :

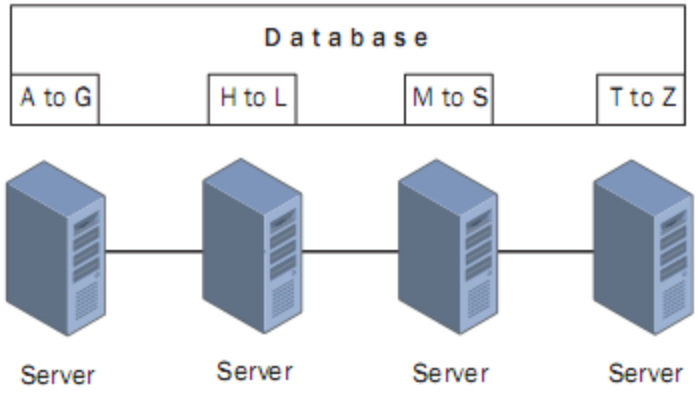
نصب برنامه های تک نمونه ای Single Instance روی سرورکلاستر راحت تر از نصب برنامه های چند نمونه ای Multiple Instance میباشد و فقط باید یک نود که همان نود فعال باشد را پیکربندی کرد و باقی نودها غیرفعال خواهند بود تا زمانی که به آنها احتیاج باشد. این نوع برنامه ها بیشتر در کلاسترهای دو نودی استفاده میشود.

برای اجرا کردن بیش از یک برنامه ی تک نمونه ای روی یک کلاستر باید چندین برنامه ی اضافه ی دیگر نیز نصب کنید. میتوان دو کلاستر دو نودی برای هر برنامه مجزا ایجاد کرد که یک نود فعال و دیگر نود غیرفعال باشد، اما با این روش همیشه دو سرور بیکار خواهند بود. میتوان یک کلاستر سه نودی ایجاد کرد که دو نود فعال داشته باشد که هر کدام یک نمونه از برنامه را اجرا میکند و یک نود غیرفعال که به حالت Standby باشد تا زمانی که به آن احتیاج باشد. اگر اینگونه کلاستر را پیکربندی کنید در حالتی که هر دو نود فعال از سرویس خارج شده باشند، سرور غیرفعال باید قادر به اجرای دو برنامه در یک لحظه باشد.

به کار بردن برنامه های Multiple Instance :

در یک برنامه ی چند نمونه ای، بیش از یک برنامه در یک کلاستر میتواند همان برنامه را در یک لحظه اجرا کند. هنگامی که برنامه های چند نمونه ای را به کار میبرید میتوانید آنها را به بخشهای دیگر تقسیم کنید و یا کپی کنید. اما کپی کردن برنامه زیاد معمول نیست و سرورهایی که از این روش استفاده میکنند معمولاً برنامه های stateless را اجرا مینمایند و این روش بیشتر برای کلاستر NLB به کار میرود.

قسمت قسمت کردن یک برنامه به این معنی است که عملکرد برنامه را به نمونه های مختلفی تقسیم کنید که هر قسمت روی یک نود کلاستر اجرا شود. به عنوان مثال میتوانید یک بانک اطلاعاتی را طوری پیکربندی و تنظیم کنید که روی یک کلاستر با ۴ نود طوری اجرا شود که هر نود اطلاعات مربوط به یک چهارم بانک اطلاعاتی را به عهده داشته باشد. همانطور که مشخص است اگر یک برنامه بتواند تعدادی از سرویسهای مختلف را در اختیار قرار دهد با این روش میتوان هر سرویس را به عهده ی یک نود قرار داد.



در برنامه های چند نمونه ای، برای توزیع درخواست ها به نودهای متناسب آن با توجه به تقسیم وظایفی که بین نودها انجام شده است مکانیزم خاصی نیاز است. هنگامی که درخواست توسط نودها پاسخ داده میشود هم باید تمامی درخواست از نودهای مختلف دوباره بازسازی و assemble شود. این مکانیزم توسط سازندگان برنامه انجام میشود و عملکردی نیست که خود کلاستر آن را انجام دهد.

عمل قسمت کردن برنامه به خودی خود میتواند باعث افزایش کارایی برنامه شود اما نمیتوانید قابلیت دستیابی بالا High Availability را هم انجام دهد. خطا در یک نود یعنی از کار افتادن یک بخش برنامه یا یک سرویس از برنامه. علاوه بر قسمت قسمت کردن برنامه باید تنظیماتی برای گذراندن وضعیت خرابی نودها نیز تنظیم شود. برای مثال در بانک اطلاعاتی که به چهار نود تقسیم شده است که در بالا اشاره شد میتوان اینطور تنظیم کرد که اگر یک نود از سرویس خارج شد نود دیگری در کلاستر وظایف آن را نیز انجام دهد البته همچنین میتوانید یک یا چندین نود غیرفعال را هم در کلاستر به صورت Standby داشته باشید که در صورت از کار افتادن سرورهای فعال وظایف آنها را انجام دهند. اما این قابلیت که سرورها بتوانند چندین قسمت را در یک لحظه انجام دهند در صورتی که نودهای دیگر با خطا مواجه شوند برای تمامی نودهای کلاستر ضروری میباشد. البته همیشه باید به نیازی که برای راه اندازی کلاستر خود دارید فکر کنید و بعد اقدام کنید، اگر میخواهید که کلاستر ۴ نودی شما هنگامی که سه نود آن از سرویس خارج شد همچنان به سرویس دهی بپردازد باید از اینکه سرور چهارم شما بتواند همزمان تمامی بخش های برنامه را اجرا کند

اطمینان حاصل کرده باشید.

انتخاب یک مدل حد نصاب

هر نود در سرورکلاستر یک کپی از بانک اطلاعاتی کلاستر را در رجیستر خود ذخیره میکند. بانک اطلاعاتی کلاستر شامل مشخصات تمامی عناصر موجود در کلاستر، شامل اجزای فیزیکی از جمله سرورها، آداپتورهای شبکه و دستگاه های ذخیره یا مخازن اطلاعاتی و آبجکت های کلاستر از جمله برنامه ها و سایر منابع منطقی میباشد. هنگامی که یکی از نودهای کلاستر به هر دلیلی خاموش میشود، بانک اطلاعاتی آن کلاستر وضعیتش دیگر آپدیت نخواهد شد. و زمانی که نود دوباره آنلاین میشود یک کپی از بانک اطلاعاتی کلاستر را خواهد داشت تا دوباره به سرویس دهی ملحق شود و این کپی را از منبع حد نصاب میگرد.

یک منبع حدنصاب کلاستر شامل تمامی تنظیمات مربوط به داده هایی میباشد که کلاستر به عنوان ریکاوری نیاز دارد و این منبع درایوی میباشد که حدنصاب در آن ذخیره شده است، در ساخت یک کلاستر اولین نود باید بتواند که کنترل منبع حدنصاب را به عهده بگیرد و به این روش میتواند اطلاعات را در این منبع ذخیره کند. تنها یک سیستم میتواند کنترل منبع حدنصاب را در لحظه به عهده داشته باشد. نودهای اضافی باید قادر به دستیابی به منبع حدنصاب باشند و از این رو میتوانند در منبع حدنصاب بانکی اطلاعاتی از کلاستر و رجیستری های خود ایجاد کنند.

انتخاب مکان منبع حدنصاب از مهمترین بخش های ساخت یک کلاستر میباشد. سرورکلاسترهایی که ویندوز ۲۰۰۳ سیستم عامل آنهاست از انواع منابع حدنصاب زیر پشتیبانی میکنند :

۱. **کلاستر تک نود single node cluster :** یک کلاستر که فقط شامل یک سرور میباشد، چرا که نیازی به اشتراک گذاری منابع ذخیره شده نیست، مکان ذخیره کردن داده

های برنامه و منبع حدنصاب در درایو کامپیوتر ذخیره میشود. مهمترین دلیل از ایجاد یک کلاستر تک نود برای تست کردن و توسعه دادن کلاستر میباشد.

۲. **دستگاه کلاستر تک منبع حدنصاب Single quorum device cluster :**
کلاستر از یک منبع حدنصاب استفاده میکند که یکی از منابع ذخیره شده ی به اشتراک گذاشته شده ای میباشد که توسط تمامی نود ها قابل دسترسی میباشد. این نوع از حالت منبع حدنصاب در اغلب کلاسترها نصب میشود و مورد استفاده قرار میگیرد.

۳. **نود دسته ای کلاستر Majority Node Set Cluster :** یک نسخه ی جدا از منبع حدنصاب در هر نودِ کلاستر ذخیره میشود به علاوه که منبع حدنصاب مسئول نگهداری تمامی کپی هایی میباشد که ذخیره کرده است، Majority Node Set Cluster بیشتر مناسب سرورکلاسترهایی میباشد که از نظر جغرافیای پراکنده هستند و یا سرورکلاسترهایی که یک منبع اطلاعاتی اشتراک گذاشته شده ندارند.

PADCO - Data Center