บทที่ 2
ระบบคลัสเตอร์

การสร้างระบบลีนุกโพรเซสคลัสเตอร์จนถึงการทดสอบประสิทธิภาพของโครงงานนั้นเราประกอบเป็นสามส่วนหลักๆและส่วนย่อยๆดังนี้

- การสร้างระบบคลัสเตอร์
  - ♦ การกำหนดคุณสมบัติของฮาร์ดแวร์
  - ♦ การกำหนดคุณสมบัติของซอร์ฟแวร์
    - ▪ ทฤษฎี เน็ทเวอร์คไฟล์ซิสเตม(Network File System :NFS)
    - ▪ ทฤษฎี เน็ทเวอร์คอินฟอร์เมชั่นเซอร์วิส(Network Information Service :NIS)

2.1 การสร้างระบบคลัสเตอร์

ก่อนอื่นเลยนั้นจะต้องทำความเข้าใจก่อนว่าระบบคลัสเตอร์นี้คืออะไร ระบบคลัสเตอร์ก็คือการนำคอมพิวเตอร์พีซีจำนวนมากมาเชื่อมต่อกันซึ่งระบบคลัสเตอร์นี้สามารถนำไปใช้กับการคำนวณแบบขนานโดยที่ระบบคลัสเตอร์นี้มักจะประกอบขึ้นด้วย เครื่องบริการเครื่องหลัก(Server) จำนวนหนึ่งเครื่อง และ เครื่องลูกข่าย(client) จำนวนหนึ่งเครื่องหรือมากกว่านั้นก็ได้อาจจะเป็นจำนวนหลักร้อยจนถึงหมื่นก็ได้ โดยทั้งหมดนี้จะเชื่อมต่อกันผ่านทางอุปกรณ์เชื่อมต่อเครือข่าย โดยระบบคลัสเตอร์นี้สร้างขึ้นโดยใช้ส่วนประกอบของอุปกรณ์ฮาร์ดแวร์ที่หาซื้อขายกันทั่วไป โดยประกอบกันขึ้นเป็นเครื่องคอมพิวเตอร์พีซีที่ใช้กับ ระบบปฏิบัติการลีนุกซ์ได้ แผงวงจรอีเทอร์เน็ตมาตรฐานได้ ฮัพหรือสวิทได้ และจะไม่คำนึงถึงความแตกต่างของอุปกรณ์ของแต่ละเครื่อง โดยจะหาซอฟต์แวร์มาใช้โดยการซื้อขายหรือจะใช้บริการจากอินเทอร์เน็ต ซึ่งซอฟต์แวร์เหล่านี้ส่วนใหญ่จะเป็นซอฟต์แวร์แบบเปิด ซึ่งหาได้ง่ายจาทางอินเทอร์เน็ต คือระบบปฏิบัติการที่จะใช้ในระบบคลัสเตอร์ก็คือ ระบบปฏิบัติการ ลีนุกซ์ จะเป็นของผู้ผลิตไหนหรือเวอร์ชั่น(version)ไหนก็ได้ แต่ทางคณะผู้จัดทำได้เลือกใช้ ระบบปฏิบัติการ ลีนุกซ์ เรดแฮท เวอร์ชั่น 9.0 ซึ่งถ้าดูจากโครงสร้างของระบบคลัสเตอร์นี้ก็คือ คอมพิวเตอร์แบบขนานที่ใช้หน่วยความจำแยกกันนั่นเอง ซอฟต์แวร์ตัวต่อไปที่ต้องหามาใช้ก็คือ เมซเซจ แพสซิ่ง อินเตอร์เฟส (message passing interface) โดยเครื่องบริการเครื่องหลักจะเป็นตัวควบคุมระบบคลัสเตอร์ทั้งหมดและบริการแก่เครื่องลูกข่าย โดยเครื่องบริการเครื่องหลักจะเป็นส่วนเฝ้าคุม(console)หรือเป็นทางผ่าน(gateway)ไปยังเน็ตเวอร์คภายนอก ในระบบคลัสเตอร์ที่ใหญ่มากๆนั้นอาจจะมีมีเครื่องบริการเครื่องหลักมากกว่าหนึ่งเครื่องก็ได้ ทุกๆเครื่องในระบบคลัสเตอร์จะถูกกำหนดคุณสมบัติและควบคุมโดยเครื่องบริการเครื่องหลักและจะถูกสั่งให้เครื่องเหล่านั้นทำอะไรบ้างเท่านั้น โดยทางคณะผู้จัดทำได้กำหนดชื่อของเครื่องบริการเครื่องหลักคือ เครื่องโอแอลแอล และส่วนเครื่องลูกข่ายซึ่งมีอยู่ด้วยกัน 4 เครื่อง ก็จะมีชื่อตามลำดับดังนี้

node01,node02,node03,node04 โดยเครื่องลูกข่ายจะมีการคัดลองข้อมูลจากเครื่องโอแอลแอลไปใช้ในส่วนของ /home กับ /usr/local ผ่านทาง เน็ตเวอร์คไฟลซิสเต็ม(Network File System :NFS) และแต่ละเครื่องลูกข่ายจะสามารถเข้าถึงเครื่องโอแอลแอลและเครื่องลูกข่ายเครื่องอื่นได้ผ่านทาง รีโมทเชลล์(remote shell :rsh) ผ่านทางไฟล์ /etc/host.equiv และเครื่องโอแอลแอลจะเป็นเน็ตเวอร์คอินฟอร์เมชั่นเซอร์วิส (Network Information Service :nis) เพื่อให้บริการข้อมูลแก่ทั้งระบบ ซึ่งผู้ใช้สามารถที่จะนั่งทำงานอยู่ที่เครื่องลูกข่ายใหนก็ได้ไม่ว่าจะเป็น node01-node04 หรือจะเป็นที่เครื่องโอแอลแอลเลยก็ได้ โดยทำการล็อกอิน(Login)เข้าสู่ระบบ ป้อนรหัสผ่าน ซึ่งสำหรับผู้ใช้คนเดียวกันจะได้สภาวะแวดล้อมแบบเดียวกันไม่ว่าจะทำการเข้าสู้ระบบผ่านทางเครื่องใหนก็ตาม

2.1.1 ทฤษฎี เน็ทเวอร์คไฟล์ซิสเตม (Network File System :NFS)

เน็ตเวอร์คไฟล์ซิสเต็มคือทฤษฎีที่ใช้สำหรับทำให้เครื่องคอมพิวเตอร์ที่ต้องการแบ่งให้ใช้ไฟล์ร่วมกันข้ามเครือข่ายได้ หรืออีกความหมายหนึ่งก็คือการที่ยอมให้เครื่องคอมพิวเตอร์เครื่องอื่นสามารถที่จะเข้าถึงอุปกรณ์ที่เก็บข้อมูลบางส่วนที่เราก็หนดสิทธ์ให้ผู้ใช้ผู้นั้นสามารถเข้าไปใช้ได้

การที่จะสร้างเน็ตเวอร์คไฟล์ซิสเต็มขึ้นมานั้นจำเป็นที่จะต้องมีความรู้พื้นฐานทางด้าน TCP/IP เน็ตเวอร์ค มาบ้างเพื่อที่จะทำให้ง่ายต่อการเข้าใจวิธีการติดตั้งซึ่งทางคณะผู้จัดทำจะได้ทำการอธิบายขั้นตอนการติดตั้งต่อไปในส่วนของการออกแบบและการติดตั้ง เคอเนล(Kernels)ทั้งหมดภายหลังจากเวอร์ชั่น 2.2.18 นั้นจะสามารถรองรับการทำงานของ NFS ได้ จุดสำคัญของการทำ NFS นั้นเมื่อเกิดปัญหาคุณจะตรวจสอบจากอะไรก่อน และจะเริ่มจากจุดใหนก่อน สิ่งที่จำเป็นที่จะต้องรู้เพื่อที่จะใช้ NFS ได้นั้นก็คือต้องมี

- ควรจะทราบเวอร์ชั่นของ nfs-utils ที่คุณใช้อยู่
- ควรจะทราบเวอร์ชั่นของ เคอเนล ที่ใช้อยู่
- และควรทราบด้วยว่าลีนุกที่คุณใช้อยู่นั้นเป็นของค่ายใหน
- และในระบบของคุณที่มีระบบปฏิบัติการอื่นนั้นมีเวอร์ชั่นอะไรมั่ง

NFS นั้นจะต้องทำการติดตั้งทั้งบนเครื่องเซิร์ฟเวอร์และบนเครื่องลูกข่ายซึ่งลูกข่ายนั้นจะเข้ามาขอเข้าใช้ข้อมูลที่เก็บอยู่ในเครื่องเซิร์ฟเวอร์ได้นั้นจะต้องทำการติดตั้งทั้งบนเครื่องเซิร์ฟเวอร์และบนเครื่องไคลเอนท์ซึ่งถ้าเครื่องเซิร์ฟเวอร์ไม่พร้อมที่จะให้บริการในส่วนของ NFS นี้เครื่องลูกข่ายก็จะไม่สามารถใช้ข้อมูลส่วนนั้นในเครื่องเซิร์ฟเวอร์ได้แต่จะใช้ของเครื่องตัวเองได้ซึ่งก็เหมือนกันถ้าเครื่องไคลเอนท์ไม่พร้อมหรือไม่เปิดใช้บริการ NFS ก็จะใช้บริการของเครื่องตัวเองแทน

ส่วนที่เกี่ยวข้องในการเปิดให้บริการ NFS นั้นจะประกอบด้วยกันสองส่วนก็คือ

- ส่วนที่อยู่บนเครื่องเซิร์ฟเวอร์
- ส่วนที่อยู่บนเครื่องไคลเอนท์

2.1.1.1 ส่วนที่อยู่บนเครื่องเซิร์ฟเวอร์

- ไฟล์ /etc/exports ไฟล์นี้จะรวบรวมรายการที่เข้ามาซึ่งแต่ละรายการนั้นจะแสดงส่วนของไฟล์ที่แบ่งกันใช้และจะบ่งบอกด้วยว่าใช้อย่างไร รูปแบบของการใช้ไฟล์ /etc/exports มีรูปแบบดังตารางต่อไปนี้

directory	machine1(option11,option12)	machine2(option21,option22)

ตารางที่ 2-1 ตารางแสดงรูปแบบภายในไฟล์ /etc/exports

ซึ่งจะอธิบายส่วนต่างๆของไฟล์ /etc/exports ซึ่งมีดังนี้

Directory นั้นจะเป็นส่วนของไดเรคทอรี่ที่เราต้องการจะแชร์ ซึ่งส่วนนี้จะไม่สนใจถึงปริมาณข้อมูลว่าจำกัดปริมาณเท่าไรโดยที่เมื่อไดเรคทอรี่ที่เราแชร์ไว้แล้วไดเรคทอรี่หรือไฟล์ที่อยู่ภายใต้ทั้งหมดนั้นจะถือว่าถูกแชร์ไปกับไดเรคทอรี่นั้นด้วย

Machine1 and machine2 เป็นเครื่องไคลเอนท์ที่ต้องการที่เครื่องเซิร์ฟเวอร์ได้กำหนดว่าจะให้เข้าถึงไดเรคทอรี่ ซึ่ง machine นี้อาจจะเป็นชื่อเครื่องที่นำมาจาก โดเมนเนมเซิร์ฟเวอร์ แอดเดรส(domain name server: DNS) หรืออาจจะเป็น ไอพีแอดเดรส(IP addresses) ก็ได้

Optionxx นั้นจะเป็นตัวกำหนดว่าเครื่องไคลเอนท์ที่จะเข้าถึงข้อมูลได้มีสิทธิที่จะเข้าถึงข้อมูลกี่แบบแบบไหนบ้างซึ่งจะมีอธิบายดังนี้

- ro: จะบอกว่าไดเรคทอรี่ที่เราแชร์ไว้นั้นไคลเอนท์หรือแมชชีนเครื่องนั้นสามารถที่จะอ่านได้อย่างเดียวไม่สามารถที่จะเขียนได้ซึ่งถ้าไม่ใส่ออปชั่น(option)ลงไปเลยค่า ro นี้จะเป็นค่ามาตรฐานที่เค้ากำหนดให้อยู่แล้ว
- rw: จะบอกว่าไดเรคทอรี่ที่เราแชร์ไว้นั้นไคลเอนท์หรือแมชชีนเครื่องนั้นสามารถที่จะทำได้ทั้งอ่านข้อมูลและเขียนข้อมูล
- no_root_squash: โดยปกติแล้วไฟล์ใดๆที่ถูกเรียกจะถูกกระทำโดยผู้ใช้ที่เป็นรูท(root) บนเครื่องไคลเอนท์ ที่ซึ่งจะถูกปฏิบัติราวกับว่ามันเป็นผู้ใช้(nobody) บนเซิร์ฟเวอร์นั้น ถ้าเกิดว่าตัวเลือกนี้ถูกเลือกไว้แล้วผู้ใช้ที่เป็นรูท บนเครื่องไคลเอนท์จะมีสิทธิการเข้าถึงไฟล์บนระบบได้เหมือนกับผู้ใช้ที่เป็นรูทบนเครื่องเซิร์ฟเวอร์ซึ่งมันอาจจะเป็นปัญหาด้านความปลอดภัยที่ต้องระมัดระวังเพราะถึงแม้ว่ามันจะจำเป็นถ้าเราต้องการที่จะกระทำงานเกี่ยวกับการบริหารใดๆบนเครื่องไคลเอนท์ที่เกี่ยวข้องกับ Exported Directory แนะนำว่าเราไม่ควรจะใช้ตัวเลือกนี้ถ้าไม่มีเหตุผลที่จำเป็น

- no_subtree_check: ถ้าส่วนหนึ่งของปริมาณข้อมูลถูกส่งออกไป จะมีรูทีน(routine)หนึ่งที่เรียกว่า Subtree Checking จะตรวจสอบว่าไฟล์นั้นที่ถูกร้องขอมาจากไคลเอนท์อยู่ในส่วนของข้อมูลที่ถูกต้องหรือไม่ถ้าจะส่งปริมาณข้อมูลออกทั้งหมดเราควรจะไม่ใช้ตัวเลือกนี้เพื่อที่จะทำให้การติดต่อสื่อสารเร็วขึ้น
- sync: โดยปกติแล้วทุกๆเวอร์ชั่นจนถึงตัวล่าสุด(Version 1.11) ของคำสั่ง exportfs จะใช้ async ซึ่งเป็นการบอกเครื่องไคลเอนท์ว่าไฟล์ถูกเขียนเสร็จแล้วซึ่งมันหมายความว่าไฟล์ได้ถูกเขียนไปบน Stable Storage เมื่อ NFS จัดการเขียนบนระบบไฟล์เรียบร้อยแล้ว การทำแบบนี้อาจจะเป็นสาเหตุให้ข้อมูลเสียหายได้ถ้าเซิร์ฟเวอร์มีการรีบูทเครื่อง และตัวเลือก sync นี้จะปกป้องการเกิดแบบนี้ได้

สมมติว่าเรามีเครื่องไคลเอนท์สองเครื่องด้วยกันคือ slave1 และ slave2 ซึ่งมี IP address ดังนี้ 192.168.0.1 และ 192.168.0.2 ตามลำดับ ซึ่งถ้าเราจะทำการแชร์ส่วนของ ซอร์ฟแวร์ไบนารี่(Software Binary) กับ โฮมไดเรคทอรี่(Home Directory) เราจะสามารถกำหนดรูปแบบของไฟล์ /etc/exports ได้ดังตารางต่อไปนี้

directory	machine1(option11,option12)	machine2(option21,option22)
/usr/local	192.168.0.1(ro)	192.168.0.2(ro)
/home	192.168.0.1(rw)	192.168.0.2(rw)

ตารางที่ 2-2 ตารางแสดงตัวอย่างการกำหนดค่าภายในไฟล์ /etc/export

- การเปิดบริการต่างๆ

หลังจากที่เราทำการกำหนดสิทธิการเข้าถึงข้อมูลและตัวเลือกที่จำเป็นสำหรับแต่ละเครื่องไคลเอนท์แล้วนั้นเราจะต้องมาเปิดบริการที่จำเป็นสำหรับเครื่องเซิร์ฟเวอร์

- การเปิดบริการ Portmapper

NFS นั้นขึ้นอยู่กับ พอทแมพเปอร์ เดมอน(Portmapper Daemon) ซึ่งจะถูกเรียกว่า portmap หรือ rpc.portmap ตัวนี้จะต้องสั่งให้เปิดเป็นอันดับแรก ซึ่งโดยทั่วไปมันจะอยู่ที่ /sbin แต่บางที ก็อาจจะอยู่ใน /usr/sbin แต่โดยทั่วไปแล้ว ระบบปฏิบัติการลีนุกจากหลายๆ ค่ายนั้นจะเปิดบริการส่วนนี้ให้อยู่แล้วแต่เพื่อความแน่ใจก่อนที่เราจะทำงาน NFS เราควรจะตรวจสอบให้แน่ใจก่อนว่าเปิดแล้วหรือยังโดยใช้คำสั่งว่า (ps aux | grep portmap)

- การเปิดบริการ Daemons

NFS นั้นจะถูกดูแลโดย daemons 5 ตัวด้วยกันคือ

  - rpc.nfsd

- rpc.lockd
- rpc.statd
- rpc.mountd
- rpc.rquotad

ซึ่ง daemons ทั้ง 5 ตัวนั้นจะอยู่ในไดเรคทอรี่ /sbin หรือไม่ก็ /usr/sbin

หลังจากที่เราได้ทำการเปิดบริการที่เกี่ยวข้องทั้งหมดแล้วเราก็ควรที่จะตรวจสอบว่าบริการทั้งหมดนั้นได้ถูกเปิดจริงซึ่งเราสามารถที่จะตรวจสอบได้ด้วยคำสั่งที่ว่า (rpcinfo –p) ซึ่งผลลัพธ์ที่ออกมาก็ควรจะมีชื่อของบริการที่เราต้องการรายงานออกมาดังตัวอย่างตารางต่อไปนี้

```
 program vers proto   port
 100000    2   tcp    111  portmapper
 100000    2   udp    111  portmapper
 100011    1   udp    749  rquotad
 100011    2   udp    749  rquotad
 100005    1   udp    759  mountd
 100005    1   tcp    761  mountd
 100005    2   udp    764  mountd
 100005    2   tcp    766  mountd
 100005    3   udp    769  mountd
 100005    3   tcp    771  mountd
 100003    2   udp   2049  nfs
 100003    3   udp   2049  nfs
 300019    1   tcp    830  amd
 300019    1   udp    831  amd
 100024    1   udp    944  status
 100024    1   tcp    946  status
 100021    1   udp   1042  nlockmgr
 100021    3   udp   1042  nlockmgr
 100021    4   udp   1042  nlockmgr
 100021    1   tcp   1629  nlockmgr
 100021    3   tcp   1629  nlockmgr
 100021    4   tcp   1629  nlockmgr
```

ตารางที่ 2-3 ตารางแสดงตัวอย่างเมื่อรันคำสั่ง rpcinfo –p

2.1.1.2 ส่วนที่อยู่บนเครื่องไคล์เอนท์

- การ Mounting & Unmount Remote Directories นั้นจะแบ่งได้ดังนี้คือ
  - ♦ การ mount ทุกครั้งเมื่อต้องการใช้บริการ NFS

ทุกครั้งที่เราต้องการใช้บริการNFS เครื่องไคลเอนท์จะต้องทำการ mount directory ที่เรามีสิทธิใหม่ทุกครั้งซึ่งจะมีรูปแบบดังตารางต่อไปนี้

```
# mount     master.foo.com:/home     /mnt/home
```

ตารางที่ 2-4 ตารางแสดงตัวอย่างการใช้คำสั่ง mount

♦ การ mount โดยให้เครื่องทำการ mount ให้ตั้งแต่เริ่มเปิดเครื่อง ส่วนสำคัญของขั้นตอนนี้ก็คือ ไฟล์ /etc/fstab ซึ่งจะเป็นไฟล์ที่เครื่องใช้ตรวจสอบตั้งแต่เริ่มต้นการเปิดเครื่องว่าต้องทำการ mount ข้อมูลส่วนไหนบ้าง โดยเราต้องนำเอา คำสั่ง ที่ต้องการ mount NFS ไปใส่ไว้ในไฟล์นี้มีรูปแบบดังตารางต่อไปนี้

```
# device                  mountpoint     fs-type     options     dump fsckorder
...
master.foo.com:/home  /mnt/home       nfs          rw          0         0
...
```

ตารางที่ 2-5 ตารางแสดงตัวอย่างการกำหนดค่าในไฟล์ /etc/fstab

♦ การ unmount เพื่อเลิกการเข้าถึงข้อมูล โดยใช้คำสั่งดังนี้ (unmount directory) ซึ่ง directory นั้นคือ directory ที่ได้ทำการ mount ในก่อนหน้านี้และไม่ต้องการใช้แล้วถึงจะทำการ unmount

2.1.2 ทฤษฎี เน็ทเวอร์คอินฟอร์เมชั่นเซอร์วิส (Network Information Service :NIS)

nis คือ บริการที่จะทำให้เครื่องอื่นๆรู้ว่าเครื่องเรามีตัวตนอยู่ใน network ส่วนที่เกี่ยวข้องในการเปิดให้บริการ NIS นั้นจะประกอบด้วยกันสองส่วนก็คือ

อันดับแรกเลยเมื่อจะใช้งาน NIS จะต้องทำการรันโปรแกรม /sbin/portmap ผู้พัฒนาบางค่ายอาจจะเก็บโค้ดอยู่ใน /sbin/init.d/ หรือ /etc/rc.d/

RPC portmapper เป็น ตัวที่คอยเปลี่ยน RPC โปรแกรมใน TCP/IP โปรโตคอล ไปเป็นหมายเลขพอร์ท มันจะต้องทำคำสั่งเพื่อสร้าง RPC call ไปยัง RPC Server ของเครื่องที่ถูกเราเรียก เมื่อ RPC เซิร์ฟเวอร์ทำงาน มันจะบอกportmap ว่า หมายเลขพอร์ทอะไรที่ใช้ในการติดต่อ และ หมายเลข RPC โปรแกรมไหนที่ได้จองพอร์ทนั้นไว้ และเมื่อไคลเอนท์ที่สร้าง RPC call เพื่อส่งหมายเลขโปรแกรม มันจะติดต่อกับ portmap ของเครื่องที่ส่งไป เพื่อตัดสินว่าหมายเลขพอร์ทไหนที่จะส่งข้อมูลไป

- ส่วนที่อยู่บนเครื่องเซิร์ฟเวอร์
- ส่วนที่อยู่บนเครื่องไคลเอนท์

2.1.2.1 ส่วนที่อยู่บนเครื่องเซิร์ฟเวอร์

- Ypserv

  จะจัด DB และ Process ไว้ให้ Ypserv จะ run เฉพาะบน NIS Server พร้อมกับ NISDB เครื่องอื่นๆจะใช้ NIS Service , ซึ่งจะมีการรัน Ypbind บน Client

- Yppasswdd

  เป็น RPC ที่ Run อยู่บน Server เป็นตัวที่เก็บข้อมูล Password ของ User เมื่อ Yppasswd ของ Client ติดต่อไปยัง Server และ Client จะส่ง User กับ Pass เก่า ไปให้ Server . rpc.yppasswd ก็จะทำการเช็ก User กับ Pass ว่า Match กันหรือเปล่า

  หลังจากที่ Server Check User กับ Pass ว่าตรงกันแล้ว ตัว Rpc.passwd จะ Execute file pwupdate passwd.* และ shadow.byname ใหม่อีกครั้ง

2.1.2.2 ส่วนที่อยู่บนเครื่องไคลเอนท์

- Ypbind

  ทำหน้าที่ หา Server ของ NIS domains และ หาข้อมูลที่เกี่ยวกับ NIS. ตัว Client จะดึงข้อมูลเกี่ยวกับRpc มาจาก ypbind หรือ อ่านจาก binding file . ตัว binding file อยู่ใน directory /var/yp

  หลังจาก binding file ได้ถูกสร้างขึ้นมา ypbind จะส่ง ypproc_Domain requests ไปยังตัว NIS Server ภายในระยะเวลา 20 วินาที ถ้าไม่มีการตอบจาก NIS Server มันจะเข้าใจว่าไม่มี Server ตัวนั้น และ Ypbind จะทำการหา NIS Server ตัวใหม่ภายในระยะเวลา 15 นาที และ Ypbind จะคอย Check ว่า ถ้า ตัว NIS Server ตัวไหน ตอบสนองเร็วที่สุดจะยกให้ตัวนั้นเป็น Server ตัวใหม่ ไปเลย

2.2 ชนิดของคลัสเตอร์

โครงสร้างของระบบคลัสเตอร์ในโลกแบ่งออกได้เป็น 2 แบบ คือ ระบบคลัสเตอร์แบบปิด และระบบคลัสเตอร์แบบเปิด ซึ่งรายละเอียดของการวางระบบในแต่ละแบบมีดังนี้

2.2.1 คลัสเตอร์แบบปิด

ระบบนี้จะต่อคลัสเตอร์ผ่านเกตเวย์ที่ซ่อนทั้งระบบจากภายนอก ข้อดีก็คือ มีความปลอดภัยสูงและจะเปลืองไอพีแอดเดรสเพียงแอดเดรสเดียวเท่านั้น ข้อเสียคือ แต่ละโหนดในระบบไม่สามารถช่วยกันบริการข้อมูลกับโลกภายนอกได้ จึงเหมาะกับการใช้งานคำนวณขนาดใหญ่เท่านั้น

2.2.2 คลัสเตอร์แบบเปิด

ระบบนี้จะต่อกับเครือข่ายภายนอกโดยตรง ทำให้ผู้ใช้เข้าถึงทุกโหนดในระบบคลัสเตอร์ได้โดยตรง ข้อเสียก็คือ ความปลอดภัยจะต่ำลงมาก เพราะต้องคอยดูแลทุกเครื่องในระบบ และยังต้องการหมายเลขไอพีแอดเดรสจำนวนมาก แต่ข้อดีก็คือระบบสามารถช่วยกันบริการข้อมูลได้ จึงเหมาะกับงานบริการข่าวสารเป็นจำนวนมาก