A Ceph, mint adattároló klaszter megoldás

Előadás anyag

Networkshop 2014, Pécs

készítette: **Szalai László** ***(szalai@inf.nyme.hu)***
Major Kálmán ***(majork@gain.nyme.hu)***
NYME INGA

Előzmények

Manapság sok rendszergazda küzd a tároló oldali redundancia és sebesség problémakörével. Sokesetben a megbízható működés érdekében drága, kommerciális megoldásokat alkalmaznak, amelyek ugyan nyújthatnak stabilitást, de skálázhatóságot csak csökkentett mértékben.

A Nyugat-magyarországi Egyetem Informatikai és Gazdasági Intézetében használunk redundáns hálózati tárolókat, de igény merült fel arra is, hogy olcsó, *Opensource* eszközökkel építsünk adattárolót, amely már az elosztott adattárolás elvén működik.

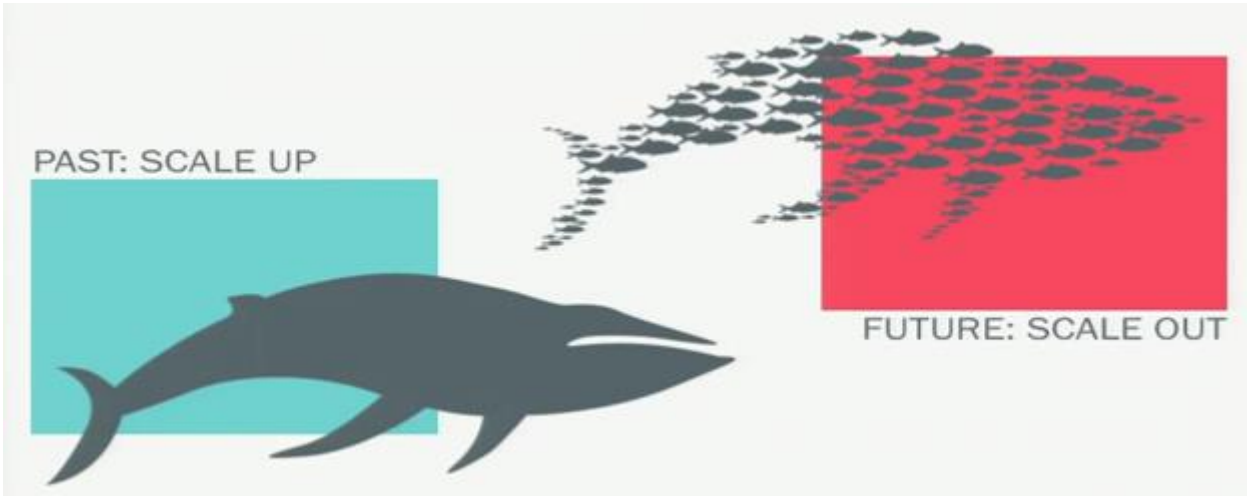
Utánanéztünk a „piacon" elérhető megoldásoknak és választottunk egy irányt, amely reményeink szerint beváltja a hozzáfűzött reményeket.

A Ceph, mint választott adattárolás

Az előrejelzések szerint 2020-ra közel 15 ZB (zetabyte) adatot fogunk tárolni. Jelenleg körülbelül 1.5 ZB adatot tárolunk.

1. ábra Adatok növekedési üteme

A manapság létező rendszerek ára folyamatosan emelkedik és egyre bonyolultabbak. Ezért még időben be kell fektetni új platformokba. Mivel egy nagyméretű adatot nehéz, lassabb mozgatni ezért egy lehetséges megoldás az, hogy kisebb darabokban többfelé másoljuk az adatainkat. Erre kínál jó megoldást a Ceph.



2. ábra Adattárolási megoldások

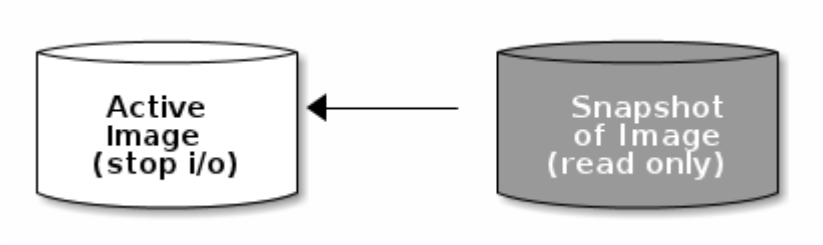
Egy jó opciónak tűnik a Ceph, mint elosztott adattároló megoldás, mellyel megismerkedtünk a félév során. Kevés olyan lehetőség volt, mely Opensource és hétköznapi számítógépekből kínál megoldást számunkra. Többek között ezért döntöttünk a Ceph mellett.

3.1 Elméleti megfontolások

A Ceph egy új tároló megoldás. Nyílt forráskódú, masszívan skálázható, elosztott objektumtároló, storage platform. Képes objektumokat, blokk eszközöket és fájlrendszert biztosítani.
A Ceph több kliensből építkezik. Nem fájlokat tárol, hanem objektumokat. Blokk eszköz lehet például egy virtuális gép merevlemeze.
A világ legnagyobb tárhely szolgáltatói például a DreamHostnál is használják, de hatékony kis- és középvállalatok kiszolgálásánál is.

3.2 Előnyök, hátrányok

A Ceph-nek nagyon sok előnye van. Nyílt forráskódú, azaz teljesen ingyenes, szemben a többi fizetős megoldásokkal. Masszívan skálázható, könnyen bővíthető, redundáns storage platform.
Kezdetektől teljesen katasztrófatűrőre tervezett rendszer, azaz a rendszer egy elemének hibája esetén nem omlik össze a teljes rendszer, így mentes a *Single Point of Failure*-tól. Mindez teljesen hétköznapi számítógépekből felállítható, egy kereskedelmi rendszernek az ára töredékéért, és annál nagyobb megbízhatósággal.
Egy fontos funkciót emelek ki a sok közül az elején, ez a snapshot.
A snapshot egy adott pontban készített írásvédett másolat az adatainkról.



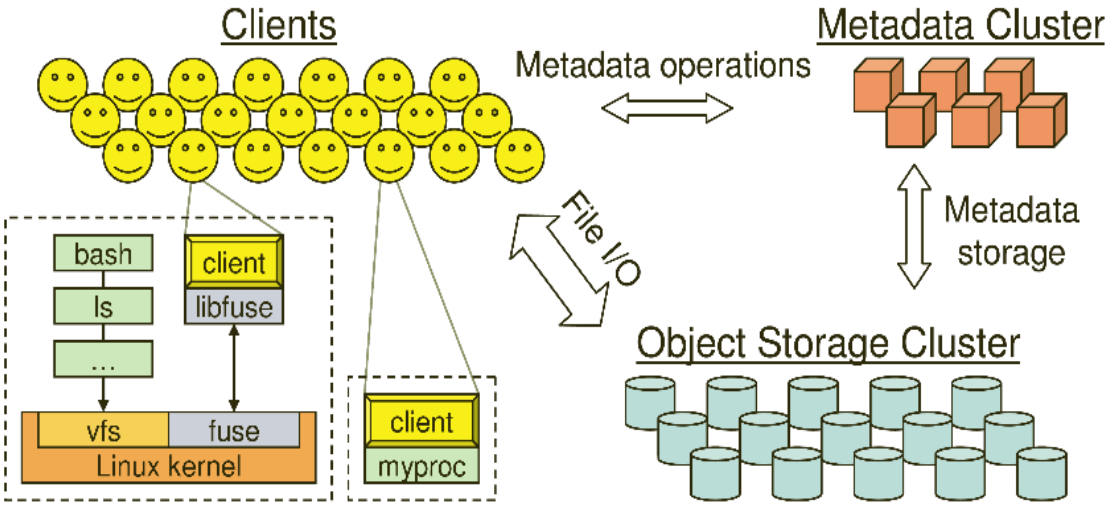
1. ábra Snapshot elvi működése

Az 1 ábrán láthatjuk a snapshot működését. Csak akkor készíthetünk pillanatnyi mentést, ha leállítjuk a klaszterünkön az I/O műveleteinket.
Csinálhatunk a blokkeszközünkről is ilyen mentést, melyeken futhatnak különböző magas szintű interfészek például többek között: QUEMU, libvirt, OpenStack, CloudStack.
Tehát a Snapshot segítségével bármikor visszatérhetünk egy adott pillanatban készített állapotra.

A Ceph úgynevezett önmenedzselt (self-managing) rendszer, ami azt jelenti, hogy képes emberi beavatkozás nélkül műveleteket végezni, így jelentősen segíti, csökkenti a rendszergazdák terhét.

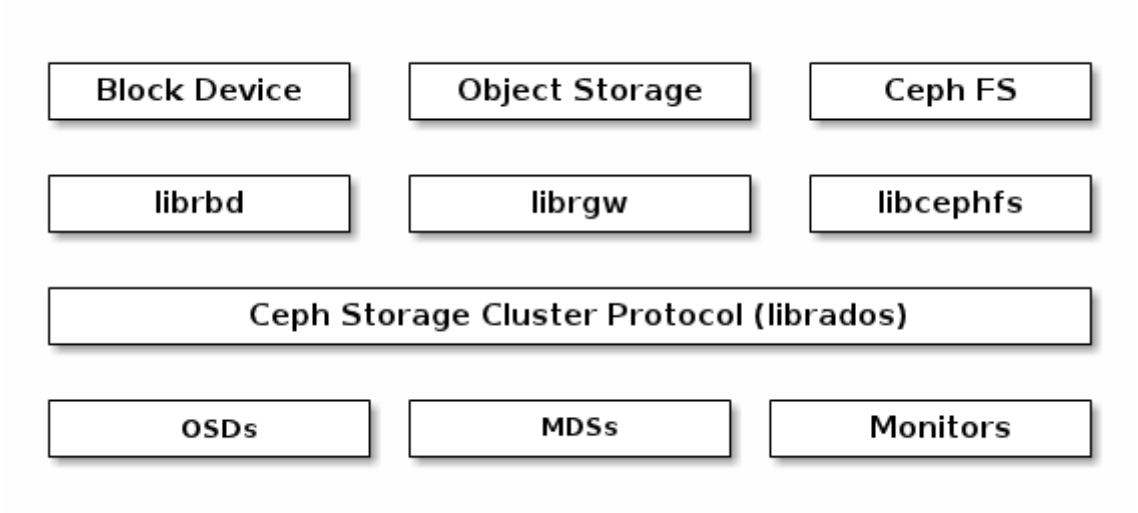
A Ceph működési elve, struktúrája

A Ceph egyedülálló módon kézbesít objektum,blokk és fájl storaget egy egyesített rendszerben. A *Ceph node* egyetlen számítógép vagy szerver a klaszterben. Egy Ceph klaszter több nodeból áll. Az alábbi ábrán láthatjuk a Ceph elvi működését.



2.ábra Ceph architektúra

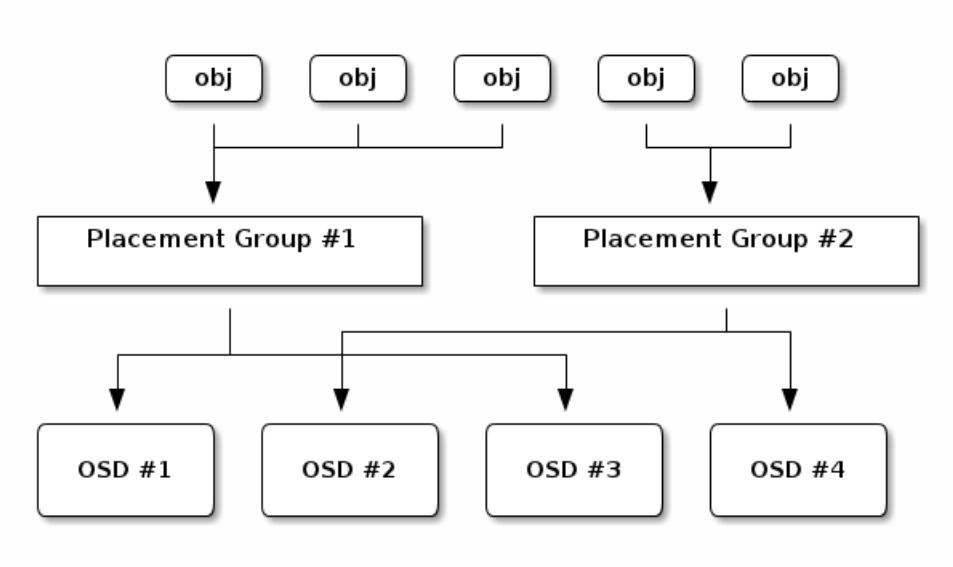
A Ceph storage klaszterek három szolgáltatásból állnak, a Ceph OSD Daemon (OSD), a Ceph Monitor (MON) és a metadata (MDS). Az OSD tárolja az adatokat objektumokként a storage nodeokon. A Ceph monitor figyeli a klaszter különböző mapjeit, beleértve a monitor map-ot,az OSD map-ot, a Placement Groupok (PG) map-eit, és a CRUSH map-ot. A 3. ábrán láthatjuk a Ceph felépítését.



3.ábra Ceph felépítése

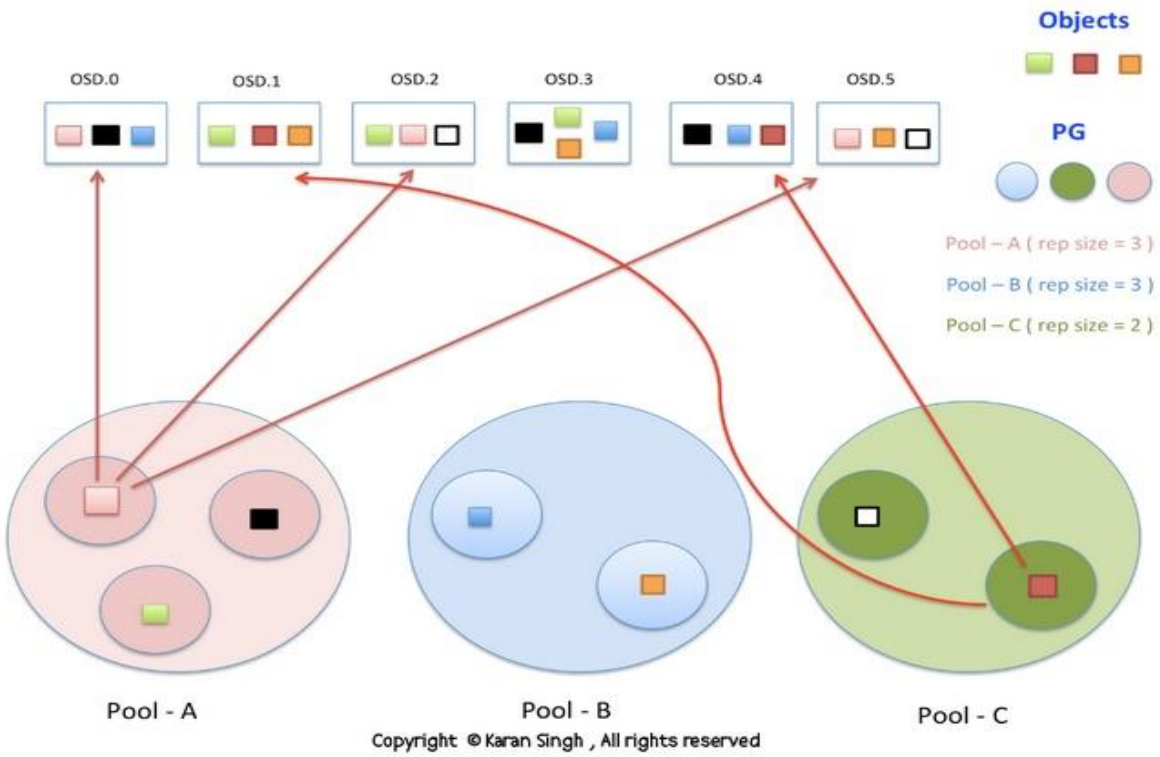
3.3.1 Objektum alapú tárolás

Az OSDknek fontos szerepük van a klaszterben. A Ceph OSD Daemon tárolja az adatokat, adat replikációkat kezel, visszaállít, visszatölti az adatokat, újra kiegyensúlyoz és ellátja a Ceph Monitorokat monitorozó információkkal, amely pedig a leellenőrzi a másik Ceph OSD daemon „egészségét” (Health-t).
Másik fontos szereplő a Placement group (PG), ami összegyűjti az objektumok sorozatát egy csoportba, és feltérképezi a csoportot OSD-k sorozatába.



4. ábra OSD-k, PG-k működése

A Ceph klaszternek vannak úgynevezett pool-jai. A pool-ok logikai csoportok az objektumok tárolására. Ezek a pool-ok készítik el a Placement Groupokat. Alapértelmezetten vannak default pool-ok (metadata,rbd,data), melyeket tudunk módosítani. Tudunk mi is létrehozni saját poo-lokat. Az alábbi ábrán láthatjuk a Poo-lok, Placement Group-ok és OSD-k közötti kapcsolatot.

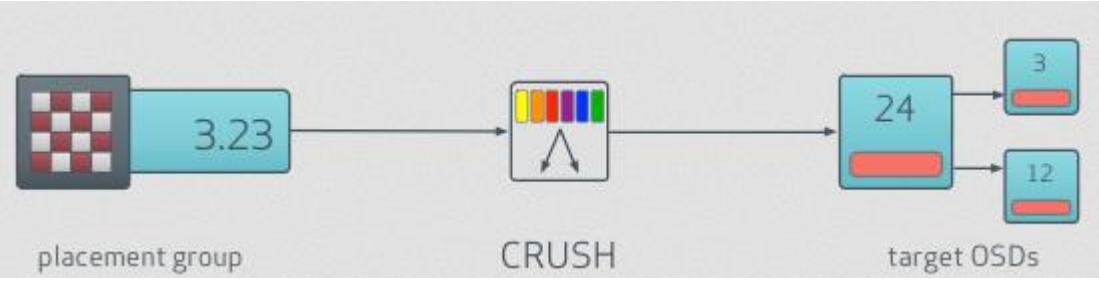


5. ábra Adatszeparáció az OSD-ken

CRUSH algoritmus

A CRUSH algoritmus eldönti, hogy hogyan és hova lehet adatokat tárolni - illetve visszanyerni - és

kiszámítja adatok helyét. A 6. ábra ezt mutatja. A CRUSH irányítja a Ceph klienseket, hogy kommunikáljanak az OSD-kkel.



6. ábra CRUSH algoritmus működése

Journaling és metadata

A Ceph OSD-k úgynevezett journal-t használnak a működésük során két okból, az egyik a sebesség, a másik a konzisztencia.
A journal engedélyezi a Ceph OSD Daemonnak, hogy írjon kisebb méretűeket, gyorsabban. Ezért szokták kihelyezni a journalt egy gyors tárolóra (SSD, SAS diszk), amely növelheti a teljesítményt.

Az MDS az egy metadata szerver daemon a Ceph fájlrendszere, amely menedzseli a fájlrendszer névterét, attribútumait és hozzáférést biztosít a megoszott OSD csoportokhoz. Mint láthatjuk, a Ceph átveheti a filerendszer menedzselés szerepét is.

Monitoring

Ha a klaszterünk nincs egy jól behatárolható lokális hálózatban, azaz ha a node-ok hálózatilag „távol" vannak egymástól, akkor célszerű úgynevezett *client.admin* titkos kulcsot generálni. Az egyes node-ok titkosított formában beszélgetnek egymással, növelve a biztonsági megfontolásokat.

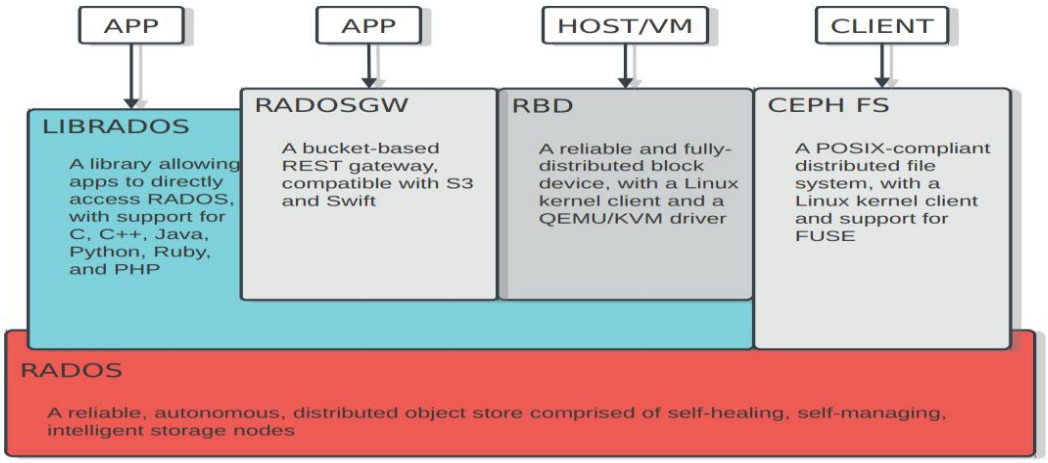
Van a node-ok között egy kitüntett gép, az *admin node*. Alapesetben ő a vezető a nodeok között.
A monitor felügyeli a gépek állapotát. Ha kiesik az egyik gép, akkor a maradék gépek quorum-on döntenek arról, hogy ki legyen a következő vezető, ehhez 50% + 1 szavazat kell. Az ábrán látható egy quorum, itt kiesett a hat gép közül egy. Szavaznak az 1,2,3,4,5 monitorok.

```
^Croot@node6:~# ceph health
HEALTH_WARN 905 pgs degraded; 617 pgs stuck unclean; recovery 282020/1659278 degraded (16.997%); 1/6 in osds are d
own; 1 mons down, quorum 1,2,3,4,5 1,2,3,4,5
```

7. ábra Quorum működése

3.4 Ceph által kínált szolgáltatások

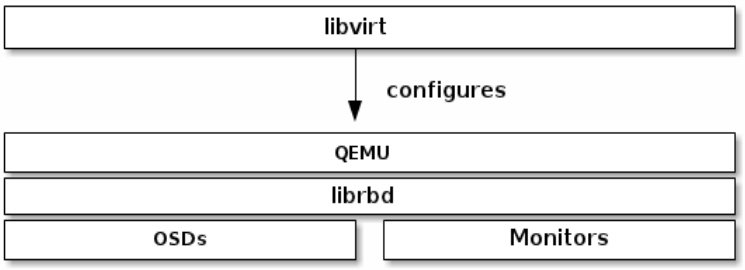
A 8.ábrán látható, milyen szolgáltatásokat kínál a Ceph. Lehetőségünk van RadosBlockDevice-t, CephFS-t, RadowGatewayt létrehozni.



8.ábra Ceph szolgátatásai

RBD

Ceph segítségével létre tudunk hozni úgynevezett RADOS Block Device-t. Ez egy blokkeszköz, amit a Linux képes kezelni kerbnelmodul és librbd segítségével.



9. ábra rbd és libvirt

Az RBD-t összetudjuk kapcsolni különféle virtualizációs megoldásokkal a libvirt segítségével.
A libvirt programkönyvtárat többféle virtualizációs technológia működtetésére használják. Az rbd-re telepíthetünk virtuális gépeket is. Így teljesen redundáns, hibatűrő virtuális gépet kapunk, ami mellesleg gyors is.

RGW

A Ceph Object Gateway egy objektum storage interfész, ami a librgw és librados könyvtárból építkezik. A RadosGW támogat két interfészt.
Az egyik az S3-compatible, ami object storage funkciókat tartalmaz egy interfésszel, ami kompatibilis az Amazon S3 RESTful API nagy alrendszerével.
A másik interfész a Swift-compatible, ami szintén object storage funkciókra építkezik egy interfésszel, ami kompatibilis az OpenStack Swift API nagy alrendszerével.
A Ceph Object Storage Ceph Object Gateway daemont használ, röviden radosgw-t, melyek FastCGI modullal interakcióban vannak a librgw és libradossal.

CephFS, Fuse

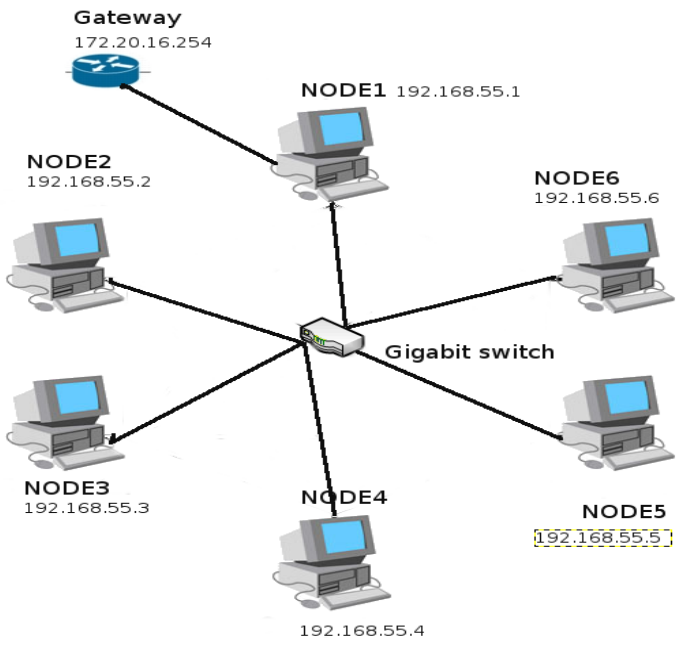
A Ceph Filesystem (Ceph FS) az egy POSIX-compliant fájlrendszer ami a Ceph Storage Clustert használja adattárolásra. A Ceph fájrendszer ugyanazt használja, mint a Ceph Storage Cluster rendszer, Ceph Block Device, Ceph Object Storage az S3 és Swift APIkkal vagy libradosszal.
A Ceph-fuse az egy FUSE client (File system USErpace) a ceph fájlrendszerhez.
Eme filerendszer kliens gépekről használható, felcsatolása után egy könyvtárat látunk, amely természetesen redundánsan kezelt a klaszter által.

Tesztkörnyezet kialakítása

Célunk volt egy olyan klaszter létrehozása, amely normál asztali számítógépekből áll és ingyenes Linux disztribúció fut rajtuk.
A klaszterünk hat darab nodeból áll. Az Ubuntu disztribúció 13.10-es verzióját választottuk.

Topológia megtervezése

Az 10. ábrán látható, hogy a klaszterünk hálózatát hogy valósítottuk meg. Mindegyik gépben egy gigabites hálózati kártya volt, a gépeket összekötöttük egy gigabites office switchhel. A klaszterünknek a 192.168.55-ös alhálózatot állítottuk be.
A Node1-es gépben (11. ábra) két darab hálózati kártya van, amelyből az egyik hálózati kártya össze van kötve a „külvilággal”, így azon keresztül érjük el távolról a klaszterünket, illetve a klaszter tagjai azon keresztül kommunikálnak az internettel. Ez utóbbi az idő szinkronizáció miatt fontos volt.

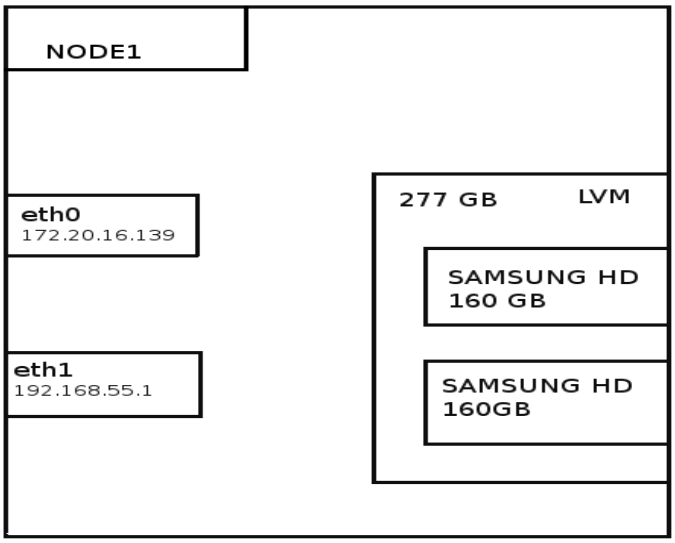


10.ábra – A klaszter hálózati topológiája

Diszkrendszer kialakítása, sebességek tesztelése

A számítógépekben két darab 160 GB-os Samsung SATA winchester van, melyek sebességeit különböző módokon megvizsgáltuk. Ötletünk alapján a két diszk LVM tömbbe lett kapcsolva, amely stripe-ot használ az adatok írására. Teszteltük az írás, olvasási sebességeket.

Megnéztük az ext4,xfs fájlrendszerekkel és stripe size méretekkel a sebességeket, a különböző benchmark programokkal.
A legjobb eredményeket az XFS és az I=4 stripe méret produkálta.



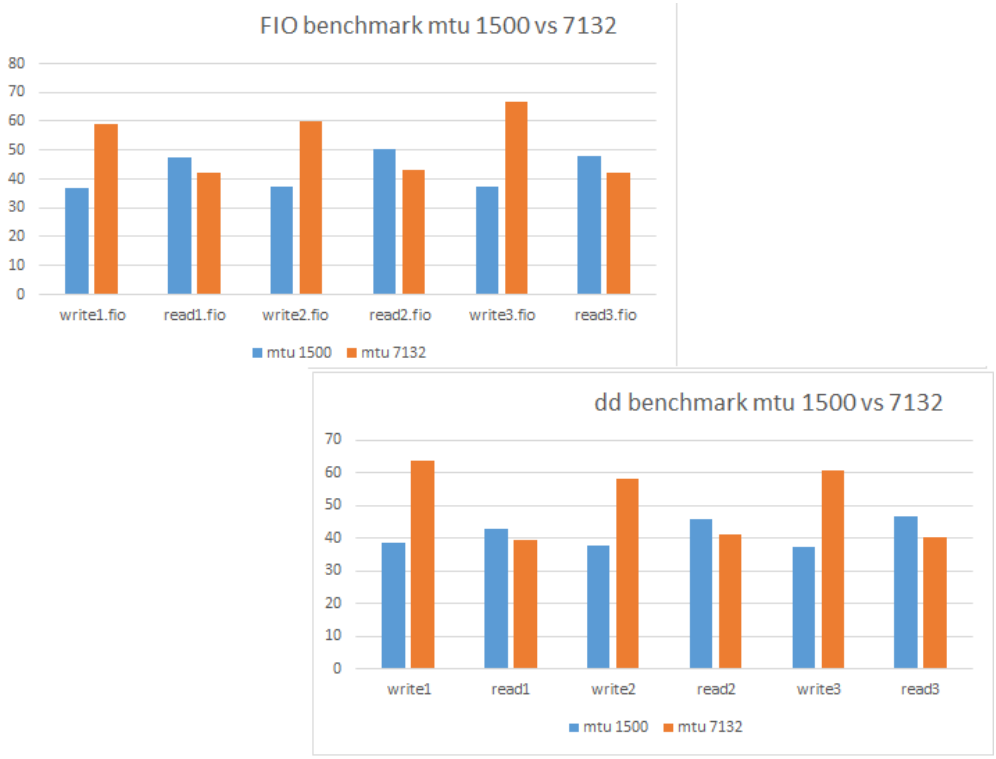
11.ábra Node1-es gép felépítése

A diszkrendszeren lokálisan elérhető volt a 80-95 Mb/mp-es olvasási és írási sebesség.

Hálózati megfontolások, sebesség tesztek

A hálózati sebesség növelése érdekében próbálkoztunk különböző beállítási lehetőségekkel. A klaszterek működése szempontjából előnyös az MTU emelése, alkalmaztuk a *JumboFrame* technológiát. A klaszter saját hálózata gigabites, amely a teljesítmény szempontjából jó, redundancia szempontjából azonban még nem elégséges.

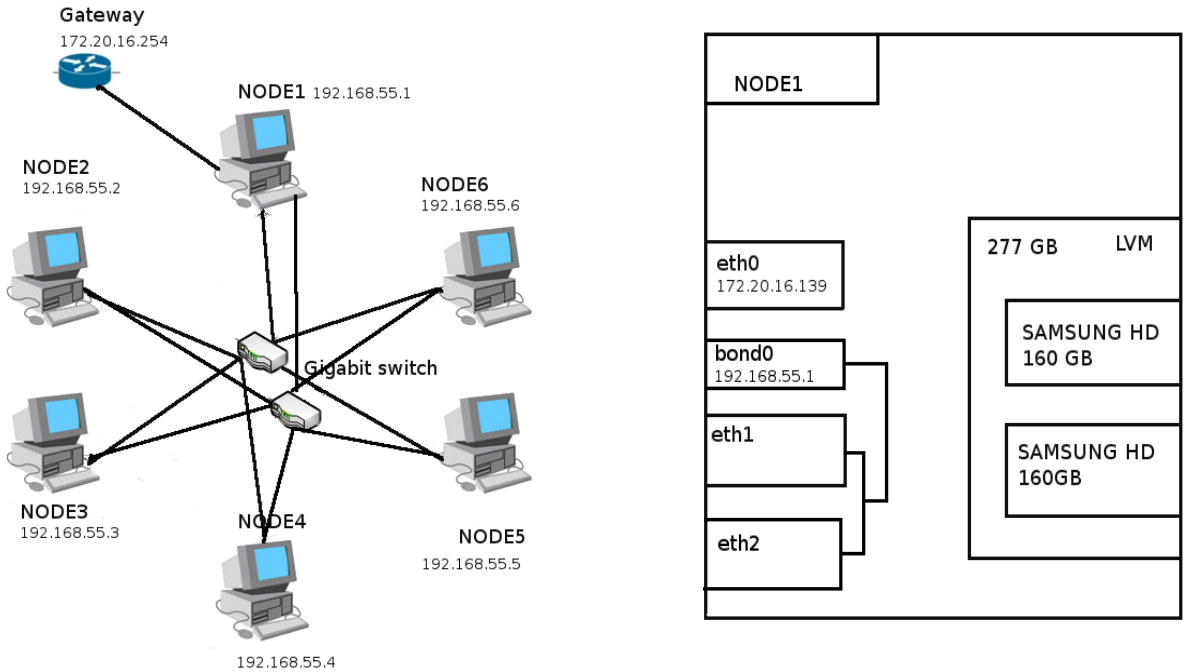
A klaszteren lefuttattunk több sebességmérést is, amely főleg a hálózati sebességekre világított rá. A klienseken futtatott méréseken látszik (13. ábra), hogy a *JumboFrame* használata jelentősen javította a sebességet. Átlagosan 52 Mb/mp-es sebesség értékeket mértünk, amely a lokális diszk sebességekhez viszonyítva 30%-os csökkenést mutat ugyan, de kaptunk érte klaszterfunkciókat.



13. ábra Klaszter írási és olvasási sebességek

Hálózati bonding alkalmazása

A redundancia növelése miatt még egy hálózati kártyát tettünk a gépekbe és egy újabb gigabites switch-el összekötöttük a node-okat. A megvalósítást mutatja a 14. ábra.



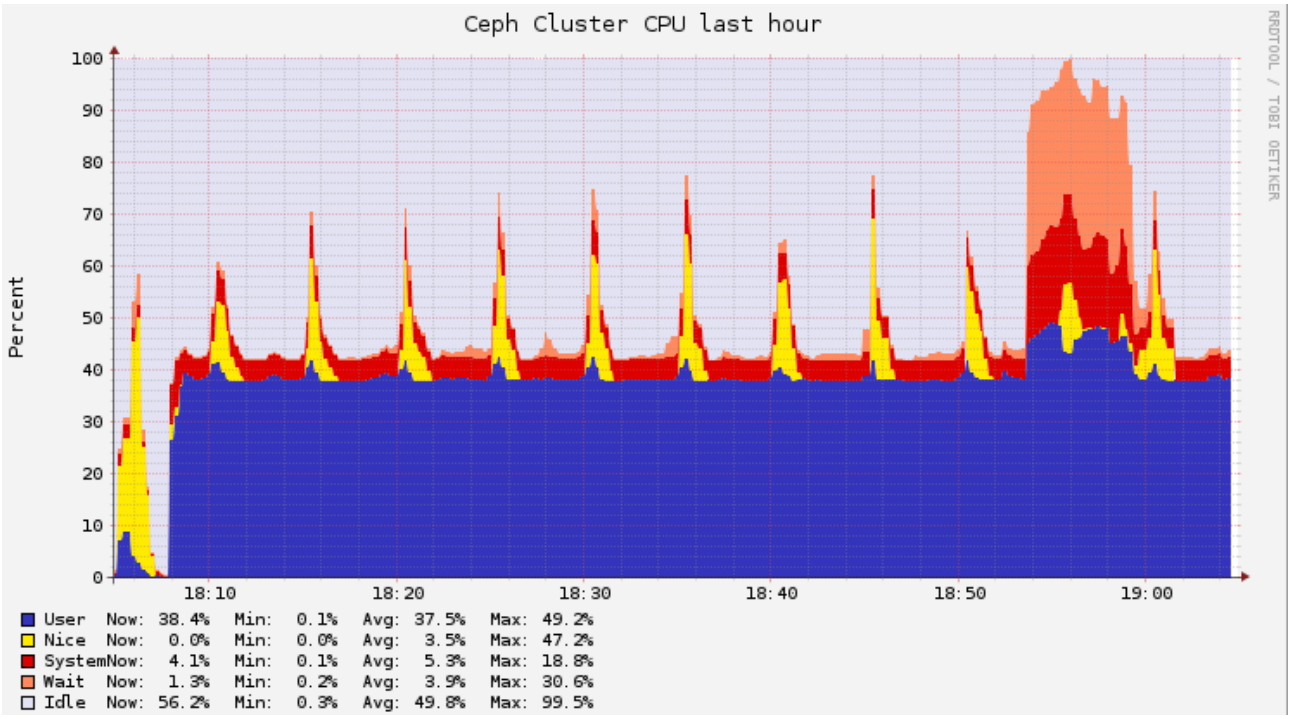
14.ábra Bonding a klaszterben

A tapasztalatok szerint a round-robin alapú bonding esetében működött a magas rendelkezésre állás, azonban a sebesség drasztikusan visszaesett. Egyéb bonding megoldások esetében pedig éppen a agas rendelkezésre állást is elveszítettük a teljesítmény mellett. Ennek oka az office switch-ekben keresendő, ugyanis ezek nem támogatják a link aggregáló megoldásokat.

Működés monitorozási megoldások

Klaszterünk működését és teljesítményviszonyait monitoroztuk is külső programmal. Egy lehetséges megoldás monitorozásra a *Ganglia* nevű szoftver.

Egy külső kliensen alkalmaztuk az RBD megoldást majd megnéztük a grafikonokat. A 15. ábrán látható egy grafikon, amely mutatja, nagy file-ok írása közben a klaszter CPU terhelése nő. Megfigyelhetőek csúcsok is, ennek magyarázata a klaszter cache kezelésében keresendő.



14. ábra A klaszter működés közben

A hálózati grafikonok azt mutatták, hogy az egyes node-okon különböző mértékben nőtt a forgalom, amely a CRUSH algoritmus és az adat replikáció-szám értékeivel van összefüggésben.

Konklúzió

Építettünk aránylag olcsó számítógépekből és hálózati eszközökből egy Ceph klasztert, amely tesztüzemben hozta az elvárásainkat. Képes volt 40-70 Mb/mp adat írási és olvasási sebességre, kezelte a magas rendelkezésre állást, a snapshot funkciót és be tudtuk illeszteni szolgáltatásait jelenlegi rendszereinkbe, mind file szinten, mint blokkeszköz szinten.

Virtuális környezetek alá is ideális, a nagyobb hypervisor-ok könnyedén kezelik a Ceph szolgálatásait.

Összességében jó benyomást keltett bennünk a termék, a jövőben részletesebb tesztekek futtatunk rajta, amely várhatóan megerősít minket abban, hogy produktív üzemben is használjuk majd.