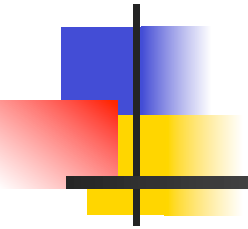
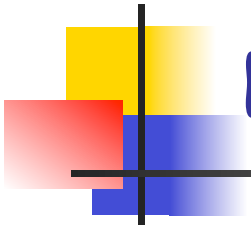


Monitoring

Alessandro Brunengo INFN-Genova

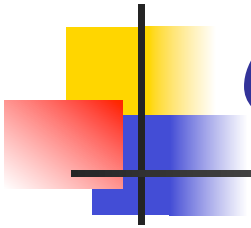


mmpmon

- Utility per raccogliere dati relativi all'I/O di GPFS
- Puo' raccogliere:
  - dati di I/O eseguiti dal nodo per file system
  - dati di I/O eseguiti dal nodo globalmente
  - istogramma dei dati di I/O selezionati per size range e, all'interno di ogni range, per intervallo di latenza
- Puo' essere utilizzato per raccogliere la statistica anche su altri nodi del cluster
  - solo nodi appartenenti allo stesso cluster

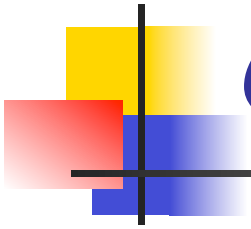
Utilizzo di mmpmon

- usage:
  mmpmon -i <infile> -r <repeat> -d <delay-in-ms> -t <timeout-in-s> -p
  - mmpmon legge una sequenza di comandi da input file (o stdin)
  - li esegue
  - aspetta <delay> ms (default 1000)
  - li riesegue, fino a <repeat> volte (default: 1)
  - -p: scrive un output machine-readable
  - -t: tempo massimo di attesa della risposta da daemon gpfs prima di tornare errore



Comandi

- il file di input contiene una sequenza di comandi, uno per linea
  - **fs_io_s**: stampa i contatori di I/O relativi a ciascun file system montato sul nodo
    - read,write,byte read,byte write,open, close, readdir,inode-update
  - **io_s**: stampa i contatori di I/O complessivi del nodo (come per fs_io_s)
  - **reset**: esegue un reset dei contatori
    - i contatori vengono resettati ad ogni umount

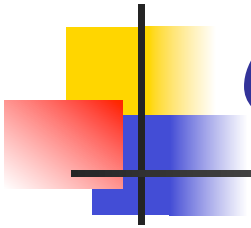


Comandi

- I comandi vengono eseguiti sul nodo locale
- E' possibile definire un elenco di nodi su cui vengono eseguiti i comandi successivi:
  - **nlist new <node> [<node> ...]**: crea la lista
  - **nlist add <node> [<node> ...]**
  - **nlist sub <node> [<node> ...]**
  - **nlist s**: stampa la lista dei nodi
  - **nlist del**: rimuove la lista
- ***** significa tutti i nodi del cluster, **.** significa il nodo su cui e' eseguito il comando

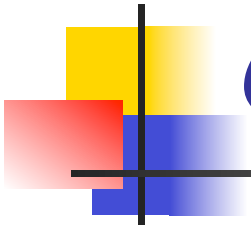
Comandi

- Gli istogrammi vengono riempiti con serie di contatori di numero di read e di write, separati per dimensione e latenza
- Per definire gli intervalli:
  - **rhist nr <size-string> <latency-string>**
    - le stringhe sono costituite da numeri che definiscono i limiti degli intervalli (da 0 al primo, dal primo al secondo, ..., dall'ultimo in su) nseparati da ";".
    - per la dimensione, sono interi che rappresentano i byte (accetta i moltiplicatori k, m)
    - per la latenza, sono numeri decimali che indicano i millisecondi
    - "=" significa "**immutata**", "*****" significa "**default**"



Comandi

- **rhist on**
- **rhist off**
- **rhist reset**
  - unico modo per resettare i contatori
- **rhist p**: stampa la corrente definizione degli intervalli
- **rhist s**: stampa i valori dei contatori



Comandi

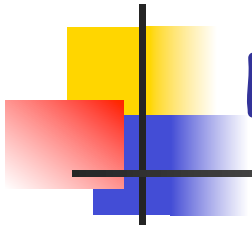
- **ver**: stampa la versione
- **once <comando>**: esegue il comando solo nella prima ripetizione
- **source <file>**: esegue i comandi che trova sul file specificato, e prosegue

Esempio

- # cat mmpmon.inp
  nlist new gpfs3 gpfs4
  io_s
- # mmpmon -p -i mmpmon.inp
  nlist _n_ 193.206.150.16 _nn_ gpfs3 _req_ new _rc_ 0 _t_ 1323822339 _tu_ 802303 _c_ 2
  nlist _n_ 193.206.150.16 _nn_ gpfs3 _req_ new _rc_ 0 _t_ 1323822339 _tu_ 802303 _ni_ gpfs3 _nx_ gpfs3 _nxip_ 193.206.150.16
  nlist _n_ 193.206.150.16 _nn_ gpfs3 _req_ new _rc_ 0 _t_ 1323822339 _tu_ 802303 _ni_ gpfs4 _nx_ gpfs4 _nxip_ 193.206.150.17
  nlist _n_ 193.206.150.16 _nn_ gpfs3 _req_ new _rc_ 0 _t_ 1323822339 _tu_ 802303 _did_ 2 _nlc_ 2
  _io_s_ _n_ 193.206.150.16 _nn_ gpfs3 _rc_ 0 _t_ 1323822339 _tu_ 804418 _br_ 241171617811 _bw_ 0 _oc_ 24940790 _cc_ 24940789 _rdc_ 20760384 _wc_ 0 _dir_ 18567571 _iu_ 411896
  _io_s_ _n_ 193.206.150.17 _nn_ gpfs4 _rc_ 0 _t_ 1323822339 _tu_ 804374 _br_ 89448744332 _bw_ 37278973952 _oc_ 11283 _cc_ 11282 _rdc_ 2447906 _wc_ 35552 _dir_ 6137 _iu_ 447

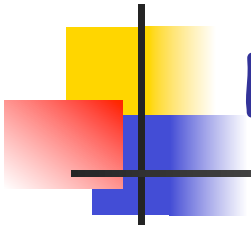
Utilizzo di mmpmon

- Utility che permette di collezionare dati per fare statistica su
  - I/O per file system, per nodo e globale sul cluster
  - correlazione di I/O
    - per dimensione
    - per latenza
    - per nodo
- Va integrata con un buon lavoro di scripting
- Si deve valutare l'impatto sulle prestazioni del cluster (in particolare per le funzioni **rhist**)
- Vedere "Advanced Administration Guide" per esempi



nsdperf

- Utility in /usr/lpp/mmfs/samples/net
- Realizza un traffico di rete tra i nodi del cluster in modo da simulare il traffico di un cluster GPFS
  - modalita' diverse da quello che si puo' ottenere via iperf/netperf
- Da compilare:
  - g++ -O2 -o nsdperf -lpthread -lrt nsdperf.C



nsdperf

- Utilizzo:
  - si esegue in modalita' server su tutti i nodi del cluster

    ```
    # mmdsh 'nsdperf -s </dev/null >/dev/null 2>&1 &'
    ```

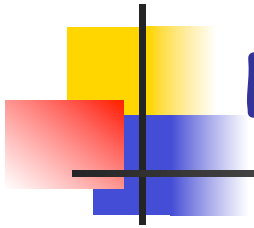
  - si esegue in modalita' interattiva su un nodo di management della sessione
  - si eseguono comandi

nsdperf: Esempio

- **# ./nsdperf**
  nsdperf> server gpfs4
  Connected to gpfs4
  nsdperf> client totem01 totem02
  Connected to totem01
  Connected to totem02
  nsdperf> test

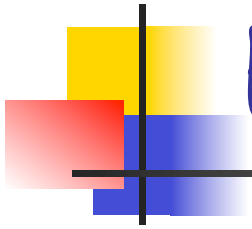
  2-1 write 235 MB/sec (56.1 msg/sec), cli 2% srv 2%, time 10, buff 4194304

  2-1 read 236 MB/sec (56.2 msg/sec), cli 5% srv 3%, time 10, buff 4194304



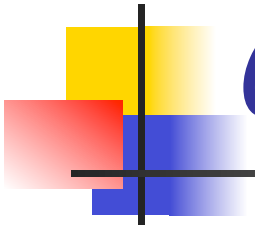
nsdperf

- E' possibile definire svariati parametri
  - test time, buffer e socket size, numero di thread, numero di socket paralleli, test type: read e write vero un server o round-robin su tutti i server definiti, bidirezionale, write in modalita' client-nsd server
- Vedere il file README



Esempi di utlity

- Altre utilities in /usr/lpp/mmfs/samples/...
  - gpfsperf: opera I/O su disco in diverse modalita' (sequenziale, stride)
    - anche questa ampiamente configurabile



GPFS SNMP

- GPFS supporta SNMP per monitorare lo stato del cluster
  - implementa un SNMP subagent per la gestione di un subtree di MIB relative a GPFS stesso
- Si possono implementare applicazioni per
  - avere una visione dello stato del cluster
  - ricevere notifiche immediate in occasione di eventi
- Un nodo del cluster deve essere configurato come snmp-agent
  - il nodo rispondera' a query snmp e potra' inviare snmp traps

GPFS/SNMP configuration

- Sul collector si installano
  - net-snmp (snmpd)
  - i MIB di GPFS (normalmente in /usr/share/snmp/mibs)
- Si deve configurare snmpd (/etc/snmp/snmpd.conf):
  - master agentx
  - AgentXSocket tcp:localhost:705
  - trap2sink <trap-destination-host>
- Opzionalmente, per cluster di grosse dimensioni:
  - agentXTimeout 60
  - agentXRetries 10

GPFS/SNMP configuration

- Si configura quindi il nodo come collector:
  - **mmchnode --snmp-agent -N node**
  - l'SNMP subagent parte automaticamente allo startup di GPFS sul nodo configurato come snmp-agent
  - il comando mmchnode forza lo start dell'SNMP subagent sul nodo designato
- Il subagent si connette al master agent (snmpd) per comunicargli la porzione di OID gestita dal subagent
- Il subagent si connette ad mmfsd per aggiornare il valore degli OID
- Gestisce una pletora di informazioni di stato del cluster, configurazione e statistica
- Supporta SNMP traps per svariati eventi
- Vedere "Advanced Administration Guide" per maggiori informazioni