Fuzzy Pool Balance:

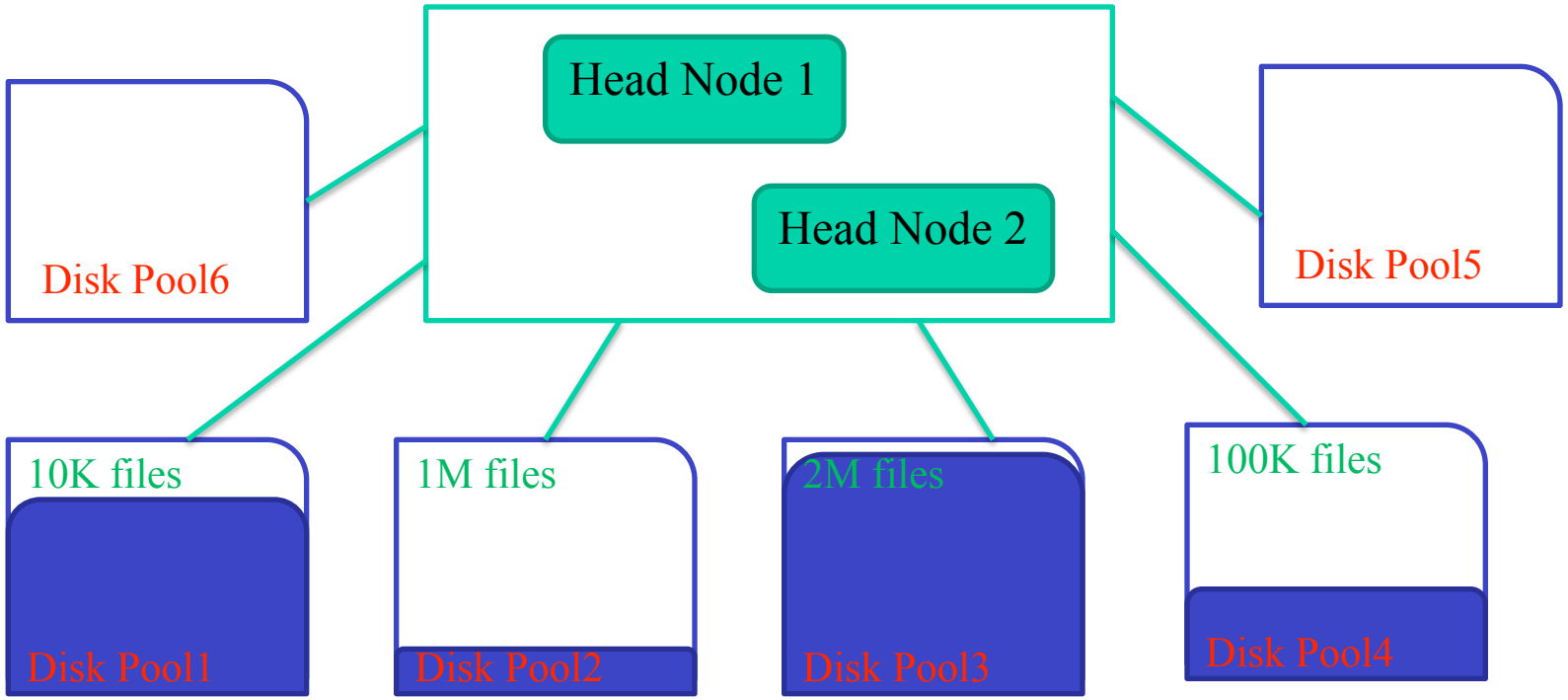
An algorithm to balance file count and available disk in distributed storage systems

Wenjing Wu

Computer Center, IHEP, China

wuwj@ihep.ac.cn

Imbalanced Disk Pools



Issues

- Cause:
  - ✓ Limits of the scheduling modules of the storage system
  - ✓ Gradual addition of new disk pools
- Two dimensional imbalance:
  - ✓ Available space among disk pools
  - ✓ File count in each disk pool

Harm to distributed storage systems

- Affects system stability, reliability and IO performance
  - Imbalanced file count among disk pools
    - Limitation from file system and OS level
    - Long initiation time for pool service
    - Requires a large amount of memory
  - Imbalanced free space among disk pools
    - Single Point of Failure
    - Reduce system throughput
    - Imbalanced system load and resource utilization
- One solution is to migrate files among disk pools to achieve a new balance

FPB (Fuzzy Pool Balance)

- A new algorithms FPB is proposed
- Initial condition: imbalanced space or file count distribution
- Input: the current file counts and free space in each disk pool
- Output: a migration plan defines what files need to be moved from what source pools to destination pools
- Goal: to achieve the two dimensional balance (file count and available space)
- Restrictions:
  1. The free space deviation among disk pools should not exceed a threshold value T_{max}
  2. file counts are balanced among disk pools
  3. Involving as less file movement as possible among disk pools

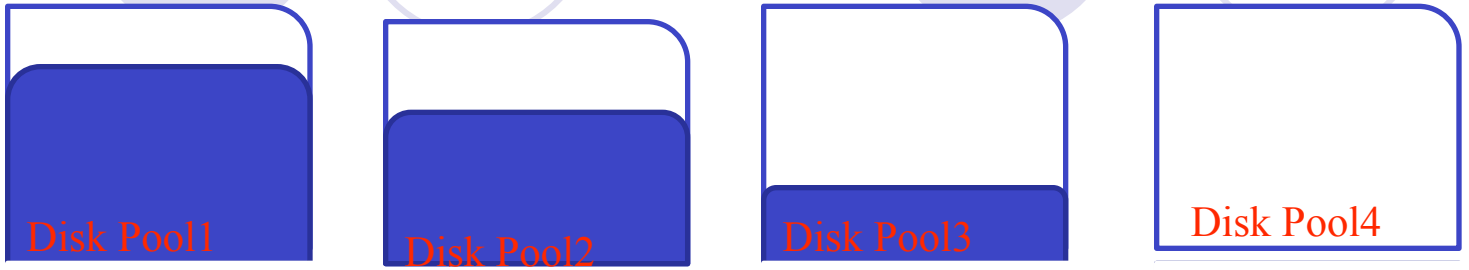
Basic Ideas (1)

- uses an array **STAT** to classify the files by their sizes:
  - Initial value such as [30K 2M 20M 2G], can be tuned into finer grain by calculation based on threshold values.
    - A file category is an interval in the array STAT, such as [2M 100M]
    - If file_size=20M, then it falls into the category of [2M 100M]
- Defines:
  - ✓ **S_{af}**: average free space of all disk pools
  - ✓ **IP (Immigration Pool)**: Disk Pool whose free space is higher than the average free space **S_{af}**
  - ✓ **EP(Emigration Pool)**: Disk Pool whose free space is lower than the average free space **S_{af}**
  - ✓ **T_{max}** : threshold value for free space deviation among disk pools

Basic Ideas (2)

- ✓ **FQR, File Quantity Ratio**, In a file category, the percentage of file count in each disk pool
- ✓ **F_{qa}**, Average **FQR** of all disk pools
- ✓ **F_{qi}, Immigration FQR,** threshold value, File Categories in IP with FQR below **F_{qi}** will immigrate
- ✓ **F_{qe}, Emigration FQR,** threshold value, File Categories in EP with FRQ above **F_{qe}** will emigrate

- The file classification array is dynamically calculated with **T_{max}** and current file size and number distribution
- Files will be migrated from EP(s) into IP(s)
- In an EP, For its desired emigration capacity, Files in its categories whose **FQR** is the higher than **Fqe** will emigrate
- In a IP, within its allowed immigration capacity quota, it accepts files in its categories whose **FQR** is lower than **F_{qi}** to immigrate.



free space: 100G | free space: 200G | free space: 300G | free space: 400G

S_{af} : (100+200+300+400)/4 = 250GB

EP : Disk Pool 1 (100G), Disk Pool 2 (200G)

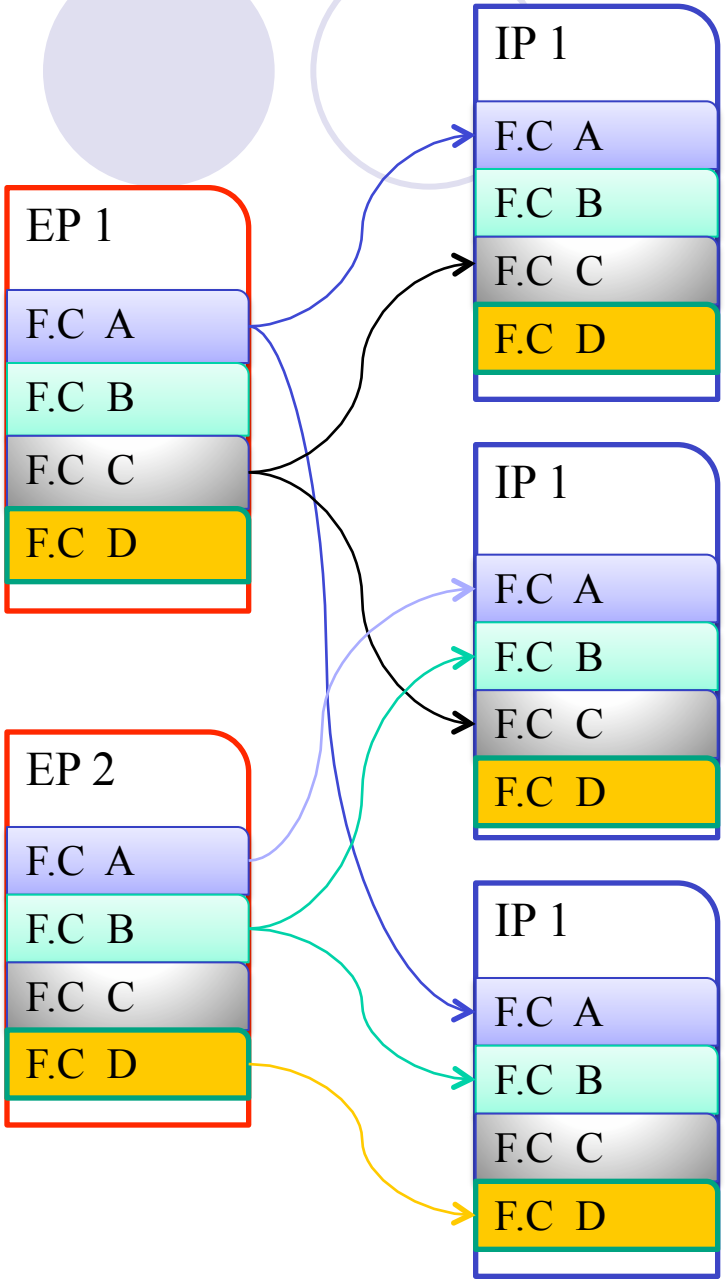
IP: Disk Pool 3 (300G), Disk Pool 4 (400G)

F_{qa} (Average FQR): ¼ = 0.25

F_{qi} : **F_{qa}** + R_{in} = 0.25 + 0.05 = 0.3

F_{qe} : **F_{qa}** + R_{out} = 0.25 - 0.1 = 0.15

R_{in} and R_{out} are correction values , - **F_{qa}** <=(R_{in} ,R_{out}) <= **F_{qa}**



- ✓ EP only emigrate files
- ✓ IP only immigrate files
- ✓ File Categories in EP(s) with a higher FQR (FQR > F_{qe}) will be emigrated
  - EP1 (A, C)
  - EP2 (A, B , D)
- ✓ File Categories in IP(s) with a lower FQR (FQR < F_{qi}) will immigrate files from EP
  - IP1 (A, C)
  - IP2 (A, B , C)
  - IP3 (A, B , D)
- ✓ Constraints:
  - In IP, FQR <= F_{qi}
  - In IP, total immigrate capacity <= maximum allowed space

FPB File Classification

- ✓ File classification is the key to achieve two dimensional balance
- ✓ Files are classified by their sizes
- ✓ Classification Array STAT is initialized with a value, then calculated into finer grain according to T_{max} and file size and number distribution
  - ➢ An initial classification array STAT [100K 2M 100M 500M 2G]
- ✓ File migration is based on the file categories
- ✓ In the same file category, files in EP(s) with FQR higher than F_{qe} will be migrated to IP(s) whose category FQR is lower than F_{qi} , so FQR tends to be evenly adjusted by migrating the capacity.

Calculation of STAT（1）

- The granularity of STAT decides the deviation between real capacity and estimated capacity of a category of files, hence decides the free space deviation among disk pools
- In interval I of STAT, namely $[St_{i-1},\ St_i]$

Define （1） $$S_{est} = \frac{(St_i + St_{i-1})}{2}$$

so （2） $$St_{i-1} - \frac{St_i + St_{i-1}}{2} \le Sf_j - S_{est} \le St_i - \frac{St_i + St_{i-1}}{2}$$

（3） $$S_{et} = S_{est} \times Nf$$

（4） $$S_{rt} = \sum_{j=1}^{Nf} Sf_j$$

so （5） $$S_{\text{var}} = | S_{et} - S_{rt} | = \sum_{j=1}^{Nf} | Sf_j - S_{est} | \le \frac{(St_i - St_{i-1}) \times Nf}{2} \le \beta$$

（6） $$\beta = \frac{T_{\max}}{M}$$

Calculation of STAT（2）

Assume **n** elements need to inserted into interval I [St_{i-1}， St_i],

（7）

$$\beta = \frac{T_{\max}}{M + n}$$

$$n \geq \frac{S_{\text{var}}}{2 \times T_{\max}} + \sqrt[2]{\frac{S_{\text{var}}^{\ 2}}{4T_{\max}^{\ 2}} + \frac{S_{\text{var}} \times M}{T_{\max}}}$$

（8.1） $$S_{\text{var}1} = \frac{(St_i - St_{i-1}) \times n_1}{2n} \leq \beta$$

（8.n） $$S_{\text{var}n} = \frac{(St_i - St_{i-1}) \times n_n}{2n} \leq \beta$$

Sum up 8.1 ~ 8.n

（9） $$\sum_{j=1}^{n} S_{\text{var}j} = \frac{(St_i - St_{i-1}) \times \sum_{j=1}^{n} nj}{2n} \leq \beta \times n$$

$$\frac{S_{\text{var}}}{n} \leq \frac{\beta \times n}{M + n}$$

$$n^2 - \frac{S_{\text{var}} \times n}{T_{\max}} - \frac{S_{\text{var}} \times M}{T_{\max}} \geq 0$$

S_{et} is the estimated size for files in interval I [St_{i-1}， St_i]

S_{rt} is the real size for files in interval I [St_{i-1}， St_i]

Sf_j is the real size for a specific in interval I [St_{i-1}， St_i]

Nf is total file count in interval I [St_{i-1}， St_i]

S_{var} is the deviation between total estimated and real size of files in interval I [St_{i-1}， St_i]

β is threshold for S_{var}

M is length of STAT

$T_{\max}$ is threshold for free space deviation among disk pools

n is number of elements to be inserted into interval I [St_{i-1}， St_i]

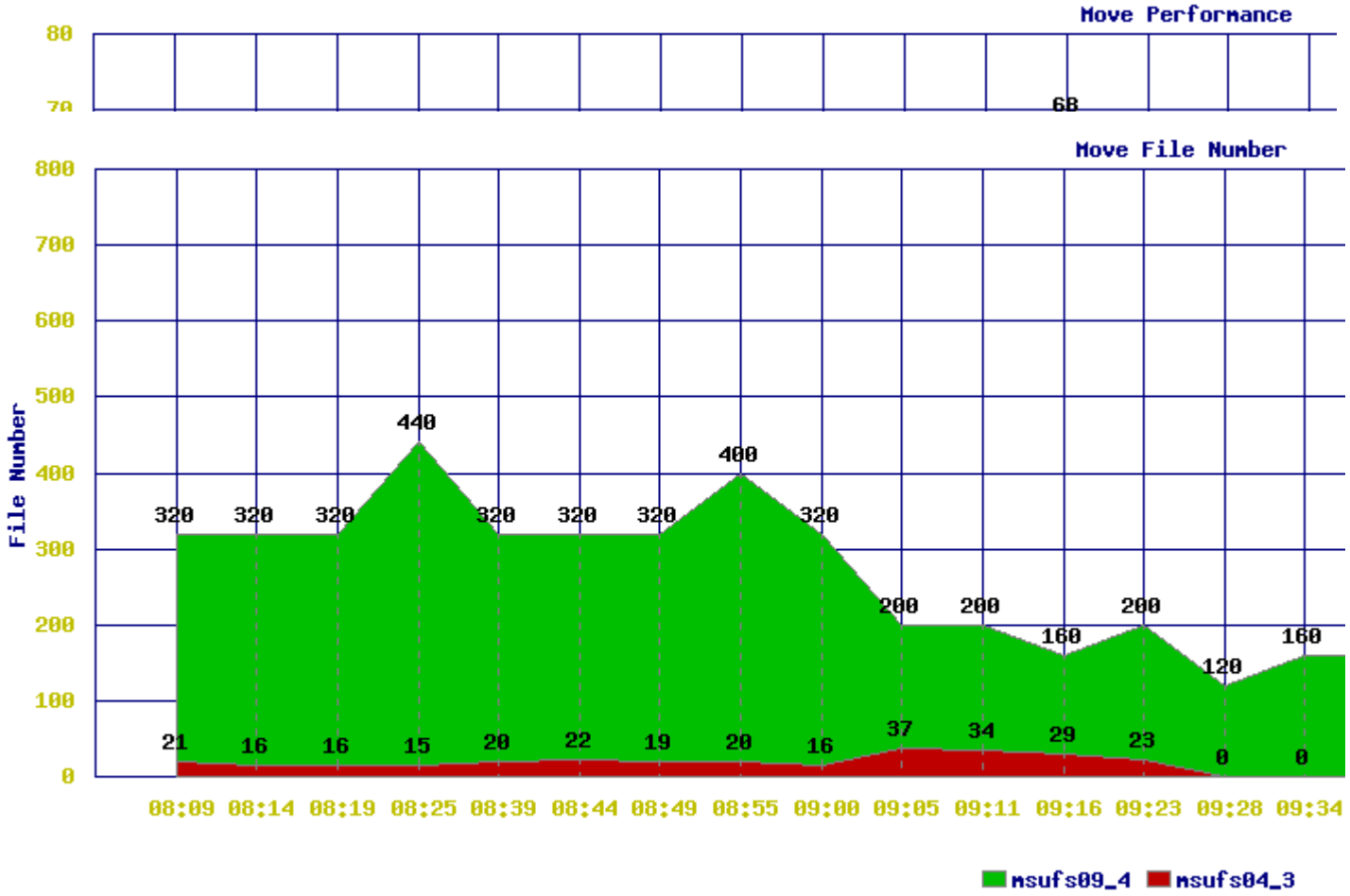
FPBM(FPB Manager):

An implementation on dCache

Basic Components:

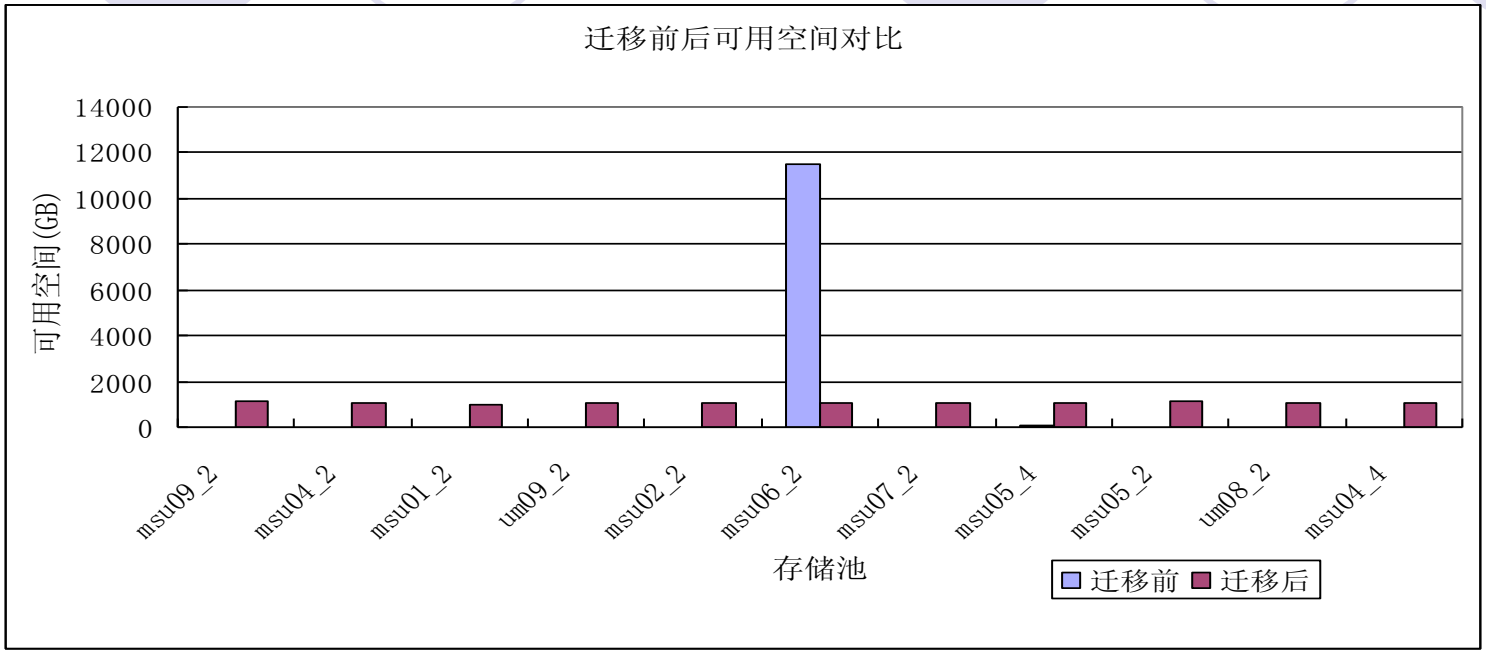
- ✓ **Classification Manager**
  - ✓ **Calculate the classification array based on Tmax value and all file sizes in all disk pools**
- ✓ **Balance Manager**
  - ✓ **Generate a migration plan based on classification array STAT and (IP, EP, F_{qe} , F_{qi})**
- ✓ **Mover**
  - ✓ **Storage system specific, move(cope/delete) files among disk pools according to the migration plan**
- ✓ **Monitor**
  - ✓ **Monitor file migrating progress, statistics**

FPBM Monitoring



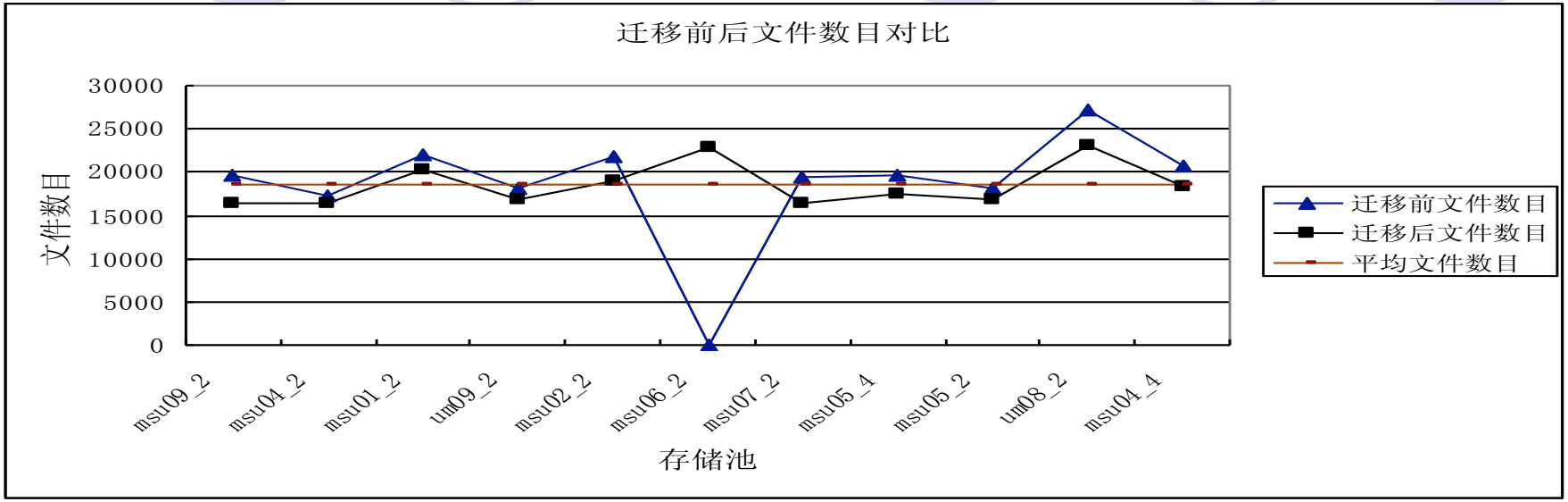
FPBM experiment on a production system:

Free space balance



- ✓ Threshold Value： T_{max} =1000GB， $F_{qe} = F_{qa}$ -0.15 ， $F_{qi} = F_{qa}$ +0.15
- ✓11 disk pools, before FPB, disk pool msufs06_2 has 11TB free space, the other 10 pools have less than 10GB space by each.
- ✓ after FPB，each pool has 1.0-1.1TB free space, free space deviation among disk pools is less than 200GB (< T_{max})

FPBM experiment on a production system:
File Quantity Ratio



File Category：[38M 44M] (files with size between 38MB and 44MB)

F_{qa} defines average FQR in all disk pools.

Before FPB, FQR is deviated from F_{qa}

After FPB, FQR is "drawn closer" to F_{qa}

In some disk pools, FQR remains the same, because of the definition of "**EP**" and "**IP**", files with a higher FQR in IP will never be emigrated, and files with a lower FQR in EP will never be immigrated

Influential Factors

Three threshold values (T_{max}, F_{qi}, F_{qe}) affect the effect of balance:

- ✓ Smaller **Tmax** results in a finer grain of classification array **STAT** which leads to better free space but worse file count distribution among disk pools
- ✓ F_{qi} and F_{qe} being closer to F_{qa} leads to better file count but worse free space distribution among disk pools

Questions & Comments?

Thanks !