# **Client-aware Cloud Storage**

Feng Chen

Computer Science & Engineering Louisiana State University

> Michael Mesnier Circuits & Systems Research Intel Labs

> Scott Hahn Circuits & Systems Research Intel Labs

#### **Cloud storage service is flourishing**



**Consumer Cloud Storage** 

- Personal cloud storage subscriptions reach **500 million** in 2012<sup>1</sup> ٠
- Public/private cloud storage market is predicted to be **\$22.6 billion** by 2015<sup>2</sup> ٠

[1] http://www.networkworld.com/news/2012/090712-personal-clouds-262231.html [2] http://www.idc.com/getdoc.jsp?containerId=prUS23097611

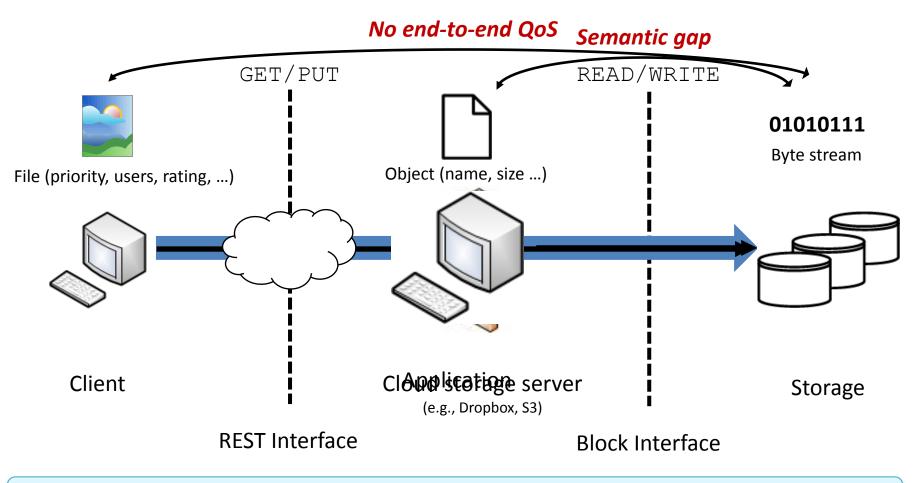
## QoS in cloud storage – a critical challenge

- QoS in today's cloud storage services
  - QoS is a real demand (e.g., Amazon Glacier, SSD-backed EC2 instance<sup>1</sup>)
  - Limitations: one-size-fits-all, coarse grained, and difficult to use
- Differentiated Storage Services [SOSP'11, VLDB'12]
  - Flexible, fine grained, and configurable to apps (e.g., database)
  - Data classification could enable end-to-end QoS in cloud storage



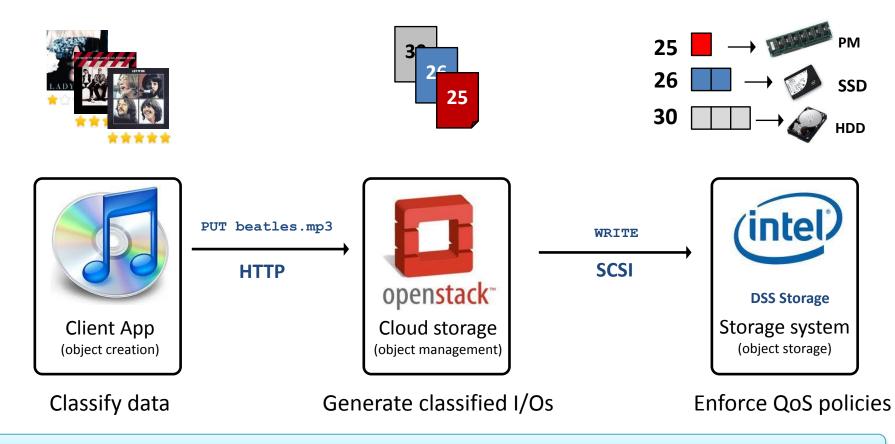
[1] http://techblog.netflix.com/2012/07/benchmarking-high-performance-io-with.html

#### Semantic gap further widens in cloud storage

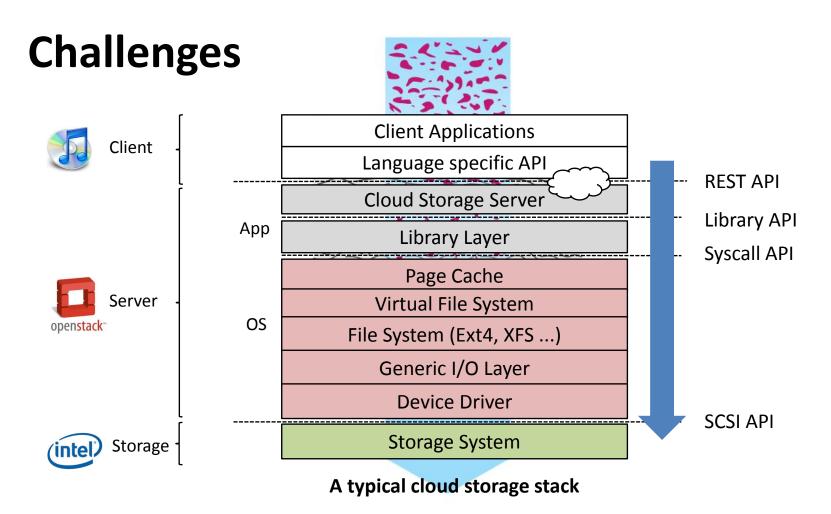


Our goal – let semantic information flow together with data from end to end

#### An example of many possible use cases ...



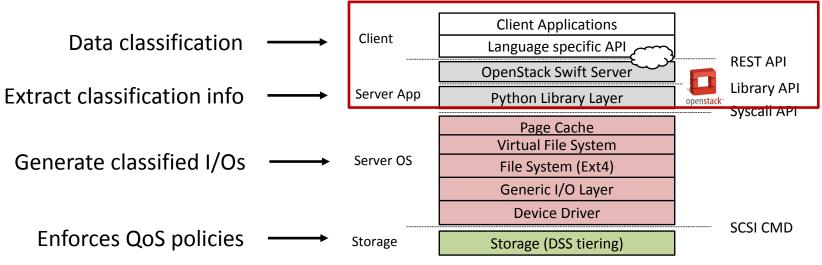
*Client-awareness* can enable many QoS opportunities (performance, reliability, security ..)



- Semantic info flow must cross multiple layers/interfaces with data
- No support in existing cloud storage for end-to-end semantic info flow

#### **Client-aware cloud storage**

- **Client** classify data and generate classified HTTP requests
- Server Modified OpenStack Swift 1.4.6 to handle classified requests
- Application API lib interface to the DSS-enabled scatter/gather I/O syscalls
- Linux OS kernel modified Linux 3.2.1 kernel with patched Ext4 to handle DSS I/Os
- Storage system Differentiated Storage Services (RAID-9) module in Linux kernel



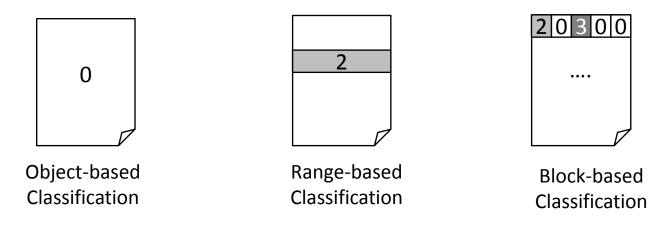
Our prototype cloud storage stack in Linux

## Outline

- Introduction
- Design & Prototype
- Experimental Evaluation
- Conclusion

### **Collecting semantic hints on client**

- Principle separating *data classification* and *policy enforcement* 
  - Similar to DiffServ in network scenario for classifying network streams
  - "What data is" (Client) vs. "How data should be treated" (Storage)
- Class a numerical value labeling a group of data (i.e., handle, index ...)
- Data classification in client-aware cloud storage
  - Object-based classification classify an entire object (e.g., music)
  - Range-based classification classify a range of data in an object (e.g., I-frames in video)
  - Block-based classification classify each block in an object (e.g., VM disk)



#### How one might classify data

- Manual classify objects in CLI, or label files in a right-click menu, etc.
- Automatic apps extract valuable semantic info (e.g., ranking for songs)
- General-purpose classify data based on file type, size, user group, etc.

						iTunes				.ax
	Clear All Categories	File Edit Controls Visualizer	Advanced	Help						
		•		•		nt To Be The Boy sining Time: 0:24		Q.		
		4					-		Search	Burn Disc
	Info	Source		Song Name	Track	Time Artist	Abum	Genre	Thy Reong	Play Count
	Info	D Lbrary	1101 4	I Want To Be The Boy	6	3:20 The White Stripes	Elephant	Roc		4
_		Radio	1102	I You've Got Her In Your Packet	7	3:39 The White Stripes	Elephant	Roc	****	2
		Music Store	1103	🖬 Ball And Biscuit	8	7:19 The White Stripes	Elephant	Roc		2
		60's Music	1104	The Hardest Button To Button	9	3:32 The White Stripes	Elephant	Roc	*****	1
	Misc	Better/Unrated Songs	1105	d Little Acoms	10	4:09 The White Stripes	Elephant	Roc		1
	initial contract of the second s	My Top Rated	1106	I Hypnotize	11	1:48 The White Stripes	Elephant	Roc		1
		Recently Played	1107	The Air Near My Fingers	12	3:40 The White Stripes	Elephant	Roc		2
- 1		Top 25 Most Played	1108	Sirl, You Have No Faith In Medicine	13	3:17 The White Stripes	Elephant	Roc		3
	Other	Top 25 Most Played	1109	It's True That We Love One Another	14	2:42 The White Stripes	Elephant	Roc	***	2
	Other		1110	Dead Leaves and the Dirty Grou	1	3:03 The White Stripes	White Blood Cells	Other	****	2
		1111 ef Hotel Yorba 2 2:10 The White Stripes White Blood Cells	Roc	****	2					
			1112	I'm Finding It Harder to Be a Gentl	3	2:54 The White Stripes	White Blood Cells	Roc	****	1
			1113	E Fell in Love With a Girl	4	1:50 The White Stripes	White Blood Cells	Roc	****	2
	Regular Meeting		1114	S Expecting	5	2:03 The White Stripes	White Blood Cells	Roc	***	2
	country and meeting		1115	d Little Room	6	0:50 The White Stripes	White Blood Cells	Roc	****	2
			<ul> <li>1116</li> </ul>	The Union Forever	7	3:26 The White Stripes	White Blood Cells	Roc.	****	1
			1117	The Same Boy You've Always Known	8	3:09 The White Stripes	White Blood Cells	Roc	****	2
	1 - Loss		1118	S We're Going to Be Friends	9	2:22 The White Stripes	White Blood Cells	Roc	****	1
	Status		1119	S Offend in Every Way	10	3:06 The White Stripes	White Blood Cells	Roc	***	1
			1120	I Think I Smell a Rat	11	2:04 The White Stripes	White Blood Cells	Other		1
			1121	duminium	12	2:19 The White Stripes	White Blood Cells	Roc	***	
			1122	🖬 I Can't Wait	13	3:38 The White Stripes	White Blood Cells	Roc	***	
- 51	ummary and Memo		1123	Mow Mary	14	1:47 The White Stripes	White Blood Cells	Roc	***	1
_	annung und mento		1124	d I Can Learn	15	3:31 The White Stripes	White Blood Cells	Other	***	1
			1125	This Protector	16	2:12 The White Stripes	White Blood Cells	Roc	10000	1
			1126	🛃 say yeah (radio)		3:22 x-session				1
	Iraant		1127	€ kernkraft 400		6:06 zomble nation	dance train 2000 -club	. Club		
	Urgent		1128	🕑 Kernkraft 400 Live remix		5:34 Zombie Nation		Tra		1
	-		1129	Moby] - Animal Rights - 07 - Soft		3:54				
			1130	dial [opus III] its a fine day		3:34				
			1131	d 01-Chemical_Brothers-Block_Rocki		5:15		1		
	UL Catagonias		1100	of an medianska's contraster	_	4.61				2.6
1	All Categories		Tot a constant		-					
_		+ 🔀 🖾			1351 s	ongs, 3.9 days, 7.01 GB			111	⑧ ❀ ▲ //

Color labeling in MS Outlook

Rating info in Apple iTunes

Client needs to be modified to send classification information to cloud storage

#### **Transmitting classifiers to server**

• An REST HTTP request (under the hood of Dropbox, S3, ...)



How to transmit classifiers to server

	In-band mode	Out-of-band mode			
Embed classifiers in <u>header</u>	<ul><li> Object-based</li><li> Range-based</li></ul>	<ul><li> Object-based</li><li> Range-based</li></ul>			
Embed classifiers in <u>object</u>	<ul><li> Object-based</li><li> Range-based</li><li> Block-based</li></ul>	<ul><li>Object-based</li><li>Range-based</li><li>Block-based</li></ul>			

#### **Embed classifiers in headers**

- Object-based classification
  - PUT with a new HTTP header "X-DSS-Object-Class"

\$ curl -X PUT -H "X-Auth-Token: abc" -T "foo" \
 -H "X-DSS-Object-Class: 25" \
 http://localhost:8080/v1/Auth\_tost/c1/foo

• Range-based classification

Class is just a handle (to lookup an associated QoS policy)

- PUT with a new HTTP header "X-DSS-Range-Class"
- Range format: <offset>-<len>-<class>
- Multiple ranges can be specified with "," in between

```
$ curl -X PUT -H "X-Auth-Token: abc" -T "foo" \
    -H "X-DSS-Range-Class: 0-64-25,1024-32-26" \
    http://localhost:8080/v1/Auth_test/c1/foo
```

Limitation – the HTTP header size is limited (e.g., 8190 bytes for Apache)

## **Embed (unlimited) classifiers in objects**

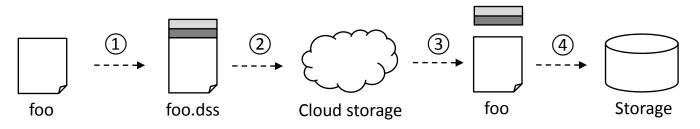
• Block-based classification

- PUT with a new HTTP header "x-DSS-Object-File: True"

```
$ curl -X PUT -H "X-Auth-Token: abc" -T "foo.dss" \
    -H "X-DSS-Object-File: True" \
    http://localhost:8080/v1/Auth_test/c1/foo
```

Instrument the object with a self-describing format

- *Metadata* interpreting the format of class section
- *Class* specifying the classes of blocks in the object
- *Data* intact content data of the object



#### An example self-describing object format

BLK\_SECTORS (4 Bytes)

NUM\_BLKS (4 Bytes)

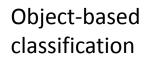
Metadata Section

CLS\_BYTES (1 Byte)

MAGIC\_SIG (2 Bytes)

VER\_ID (1 Bytes)

Block-based classification



 
 Signation
 Signation
 RESERVED (12 Bytes)
 Signation
 OBJ\_DATA (Variable)

 Metadata
 Section
 Class
 Section
 Data

 Metadata
 Section
 Class
 Class
 Section

RESERVED (4 Bytes)

BLK\_CLASS\_TABLE (NUM\_BLKS X CLS\_BYTES bytes)

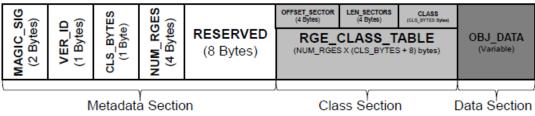
Class Section

OBJ\_DATA

(Variable)

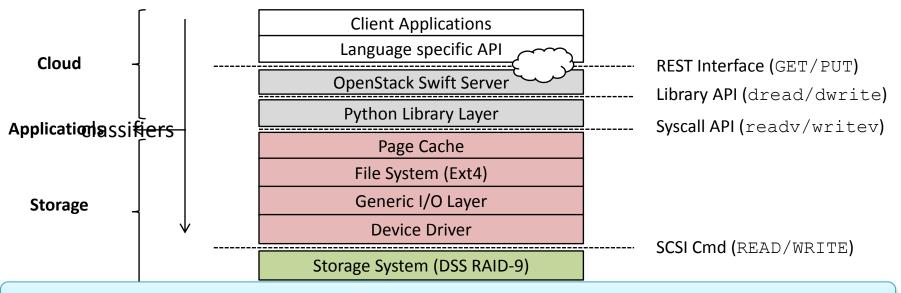
Data Section

Range-based classification



#### Handling classified requests at server

- Cloud storage server extracts classifiers from classified HTTP requests
- Generate classified I/Os using language-specific (e.g., Python) APIs
- Submit I/Os to OS kernel via scatter/gather syscalls (readv/writev)
- OS kernel receives and passes classified I/Os from app to FS, BIO, to DD
- Classifiers are copied to the 5-bit group number of READ/WRITE SCSI CDB



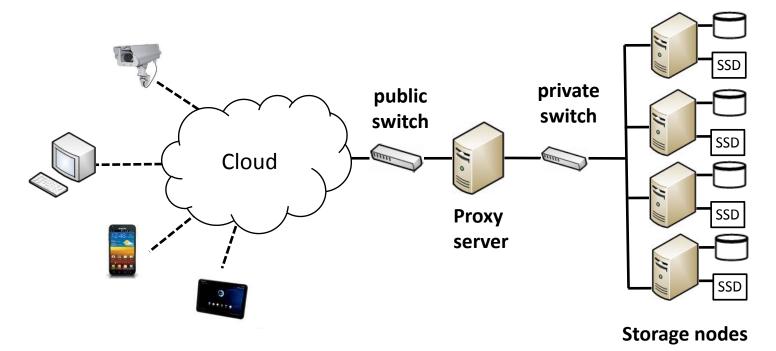
Cloud storage server needs to be modified to handle classified requests

## Outline

- Introduction
- Design & Prototype
- Experimental Evaluation
- Conclusion

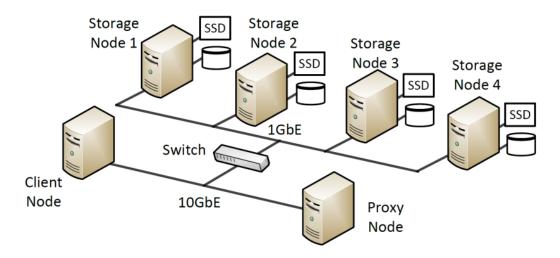
#### **Evaluation Model**

- Simulate a typical cloud storage service (e.g., Real WS)
- Compare various storage solutions for cloud storage services
  - Bandwidth, latencies, cost efficiency, etc.



#### System setup

- Cluster configuration
  - 1 Proxy & 1Client 2x 8-core Xeon Sandy Bridge 2.9GHz with 128GB memory
  - 4 Storage nodes 2x 6-core Xeon Westmere 3.3 GHz with 24GB memory
  - Storage devices 1x Intel 710 SSD, 1x Seagate Constellation 1TB SATA HDD
  - Network setup 10GbE links for proxy/clients, 1GbE links for storage nodes



A typical OpenStack Swift cluster setup

#### System setup

- Software configuration
  - OS: Fedora Core 14
  - Patched Linux kernel 3.2.1 and Ext4 for DSS
  - Modified OpenStack Swift 1.4.6 for handling classes
    - Services proxy, object, container, account, updater, replicator, auditor
    - Proxy service with 32 workers and storage services with 8 workers
    - Each storage device is set as an individual zone, 3 replicas for reliability
- Storage node configuration





**HDD-only** 

LRU caching



**DSS** caching



SSD-only

#### **Case 1: Persistent caching for cloud storage**

#### • Object classification

- Distributions are mostly based on real files (10,711 pictures, 319,073 videos)
- <u>Each object type</u> is given an individual class (USER1-USER5)
- For all object types, large files (>=10MB) are classified to USER7

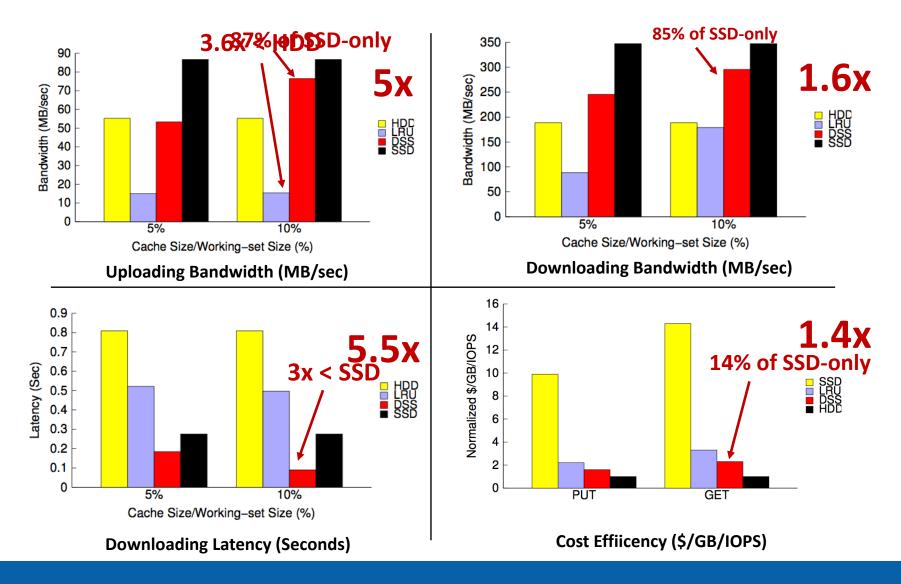
#### • DSS Caching policy

- Caching high-priority data first in the order of USER1-USER7
- LRU caching is used for data within the same class

Object Size	Files (60%)	Picture (35%)	Music (4%)	Video (0.9%)	VMDK (0.1%)	
≤64KB	USER1 (79%)	USER2 (17.1%)	USER3 (0%)	USER4 (0.3%)	USER5 (0%)	
≤512KB	USER1 (14%)	USER2 (43.6%)	USER3 (0%)	USER4 (2.8%)	USER5 (0%)	
≤1MB	USER1 (3%)	USER2 (14.5%)	USER3 (0.8%)	USER4 (4.9%)	USER5 (0%)	
≤5MB	USER1 (2%)	USER2 (20%)	USER3 (52.3%)	USER4 (32.4%)	USER5 (0%)	
≤10MB	USER1 (1%)	USER2 (3.5%)	USER3 (39.6%)	USER4 (31.7%)	USER5 (0%)	
>10MB	USER7 (0%)	USER7 (1.3%)	USER7 (7.3%)	USER7 (27.9%)	USER7 (100%)	
Distribution and Classification						

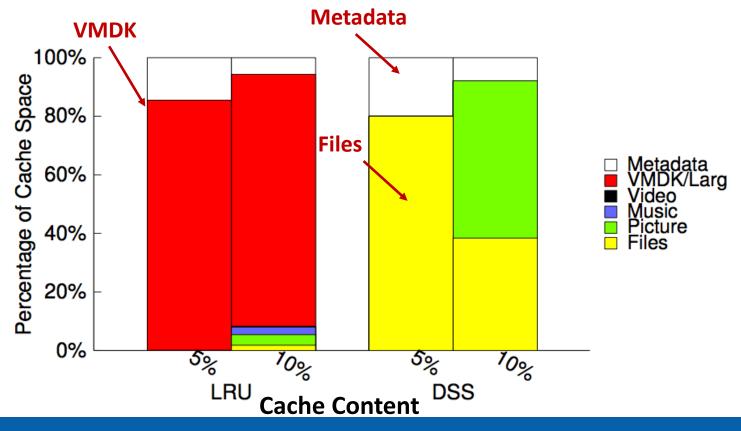
Distribution and Classification

#### **Experimental result highlights**



## **Explaining cache effects**

- CACS selectively caches the most important data
  - LRU disregards the user-specified data "importance"



### **Case 2: Fine-grained traffic control**

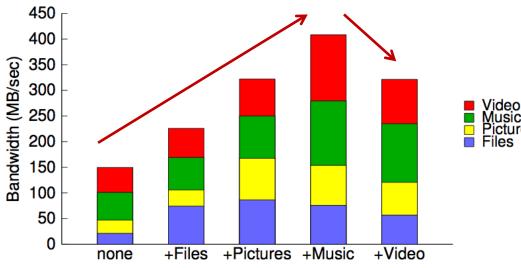
#### • Traffic direction policy

- <u>Each object type</u> is given an individual class
- Specified data objects will be directed to SSDs as needed
- The un-specified data objects will be directed to HDDs (UNCLASSIFIED)
- 25 parallel requests for each object type for uploading data
- We do four runs, and each time we add one object type to the SSD

Object Size	Files (60%)	Picture (35%)	Music (4%)	Video (1%)
≤64KB	79%	17.1%	0%	0.3%
≤512KB	14%	43.6%	0%	2.8%
≤1MB	3%	14.5%	0.8%	4.9%
≤5MB	2%	20%	52.3%	32.4%
≤10MB	1%	3.5%	39.6%	31.7%
>10MB	0%	1.3%	7.3%	27.9%

**Distribution and Classification** 

#### **Bandwidth and throughput**



#### **Directing traffic to SSD moves** pressure for workloads on HDD

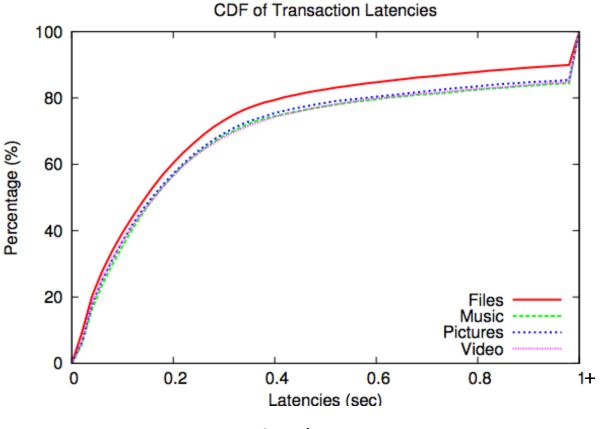
We also need to be careful parallelizing device uses is important

	Speedup					
	Files	Files Pic		Video		
+Files	3.5x	1.2x	1.1x	1.1x		
+Pictures	4x	3.1x	1.5x	1.5x		
+Music	3.6x	3x	2.3x	2.6x		
+Video	2.7x	2.4x	2.1x	1.8x		

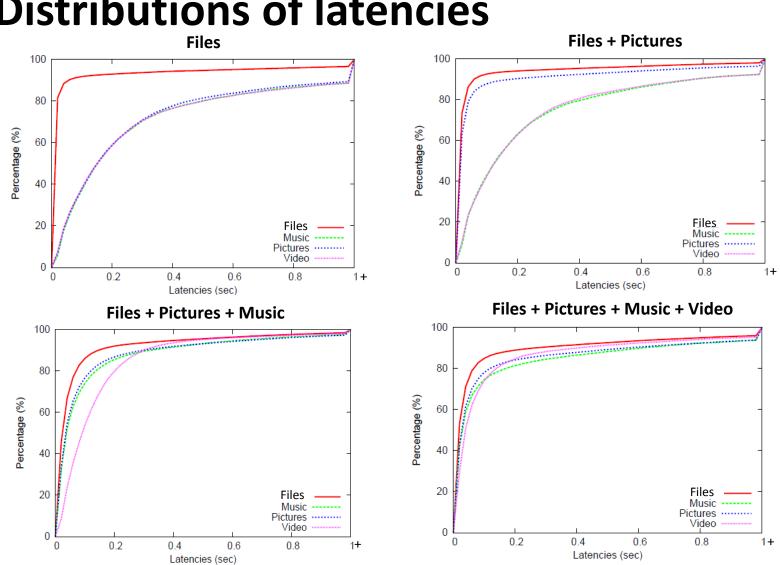
usic

ctures

#### **Distributions of latencies**



**Before I/O redirection** 



#### **Distributions of latencies**

### Conclusion

- Cloud storage brings a critical QoS challenge
  - The widening semantic gap makes end-to-end QoS difficult to realize
- A client-aware cloud storage framework
  - Client classifies data and transmits data and semantic hints to server
  - Cloud storage server handles classified requests and generates I/Os
  - Storage system enforces QoS policy based on data classification
- Client-awareness can provide significant benefits
  - 5x bandwidth, 5.5x latency, 14% of the SSD cost
  - Other optimizations are also possible (e.g., reliability, security ...)

# Thank you!

fchen@csc.lsu.edu michael.mesnier@intel.com scott.hahn@intel.com