

CS530

Scalable Wide-area Upload

[Bistro00]

Bill Cheng

<http://merlot.usc.edu/cs530-s10>



Copyright 2006, Bill Cheng

Copyright 2006, Bill Cheng

Slide

a Platform for Building Scalable Wide-Area Upload Applications



Copyright 2006, Bill Cheng

Scalable Data Transfer Applications

Subcategory: Applications

		# of Receivers	
		One	Many
# of Senders	One		
	Many		



Copyright 2006, Bill Cheng

Copyright 2006, Bill Cheng

Scalable Data Transfer Applications

Subcategory: Applications

		# of Receivers	
		One	Many
# of Senders	One	App Additional ...	
	Many		



Copyright 2006, Bill Cheng

Scalable Data Transfer Applications

Subcategory: Applications

		# of Receivers	
		One	Many
# of Senders	One	App Additional ...	we download a software distribution with on-steam and a server post ...
	Many		chat rooms video conferencing multiplayer games ...



Copyright 2006, Bill Cheng

Copyright 2006, Bill Cheng

Scalable Data Transfer Applications

Subcategory: Applications

		# of Receivers	
		One	Many
# of Senders	One	App Additional ...	we download a software distribution with on-steam and a server post ...
	Many		chat rooms video conferencing multiplayer games ...



Copyright 2006, Bill Cheng

Scalable Data Transfer Applications

Application: Application Server

# of Senders	# of Receivers	
	One	Many
One	<p>No traditional apps ...</p>	<p>web/downloads software data intensive video-on-demand server push ...</p>
Many	<p>Bistro!!</p>	<p>chat rooms video conferencing multiplayer games ...</p>

Who is Working on Uploads?

- ➡ To the best of our knowledge, there is no existing work on making many-to-one communication at the application layer **scalable** and **efficient**

What Are Upload Applications?

- ➡ **Hard deadlines**
 - IRS income tax submission
 - paper submission
 - ➡ **No hard deadlines**
 - Internet-based storage
 - Data warehousing
-

Why is Upload Different?

- ➡ many-to-one data transfer
- ➡ read vs. write
 - ➡ traditional solution such as replication of data (caching), replacement of data, etc. won't help
 - ➡ multicasting, security
- ➡ contention for service rather than data
- ➡ data consumed later (not expired data)
- ➡ replication of services and resources for a single event is expensive, inflexible, & not scalable

Scalable Data Transfer Applications

Application: Application Server

# of Senders	# of Receivers	
	One	Many
One	<p>No traditional apps ...</p>	<p>web/downloads software data intensive video-on-demand server push ...</p>
Many	<p>Bistro!!</p>	<p>chat rooms video conferencing multiplayer games ...</p>

Who is Working on Uploads?

- ➡ To the best of our knowledge, there is no existing work on making many-to-one communication at the application layer **scalable** and **efficient**

What Are Upload Applications?

- ➡ **Hard deadlines**
 - IRS income tax submission
 - paper submission
 - ➡ **No hard deadlines**
 - Internet-based storage
 - Data warehousing
-

Why is Upload Different?

- ➡ many-to-one data transfer
- ➡ read vs. write
 - ➡ traditional solution such as replication of data (caching), replacement of data, etc. won't help
 - ➡ multicasting, security
- ➡ contention for service rather than data
- ➡ data consumed later (not expired data)
- ➡ replication of services and resources for a single event is expensive, inflexible, & not scalable

Traditional Approaches

for the application server

- ➡ Increase capacity
- ➡ Spread the load ... over time, space, or both
- ➡ Change the workload
- ➡ Examples
 - ➡ data replication (ip not coding, web caching)
 - ➡ data replication (multi-resolution images, video)
 - ➡ service replication (DNS, proxies, FTP)
 - ➡ server push (new video content, software distribution)

Traditional Approaches (Cont ...)

for the client

- ➡ Routers will download histograms through data replication (caching)
- ➡ Use their own network of servers, with strategic placement of servers around the world
 - > 2000 servers
 - > 45 countries
 - > 150 networks
- ➡ Clients include Microsoft, Paramount, Wired, CBS Sports, Nike, BBC America, Apple, ...
 - ➡ Why are there local pools? availability of networks
 - ➡ real-life events

Our Goals

- A single infrastructure (named **Bistro**) for all data collection needs
 - good performance (for both service providers and users)
 - scalable (shares resources among all service providers)
 - secure (one service provider does not have to trust another)



Copyright © 2008, Spring 2008

Current State of Affairs for Uploading

- Independent data transfers over the Internet, i.e., TCP/IP
- TCP/IP shares bandwidth fairly
- Individual clients experience poor performance when number of clients is large (if transfer time is long enough to see other connections)
- TCP/IP is here to stay

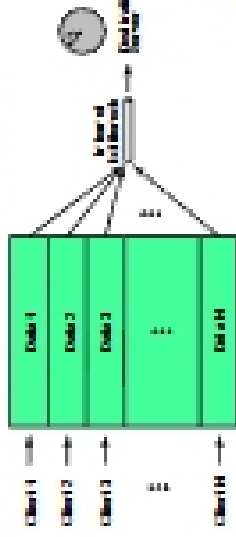


Copyright © 2008, Spring 2008

Key Observations

(Application with deadlines)

- Existence of hot spots in uploads is largely due to approaching deadlines
- Exacerbated by long transfer times
- Problem becomes much worse as little time ...



Copyright © 2008, Spring 2008

Key Observations (Cont...)

(Application with deadlines)

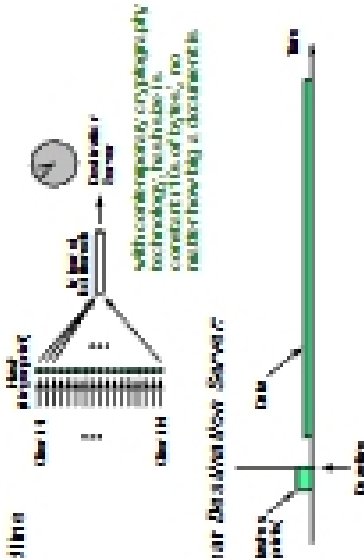
- What is actually needed is an assurance that specific data was submitted before a specific time
 - i.e., we need a commitment of what and when a submission took place
- Then the transfer of that data needs to be done in a timely manner, but does not have to occur by the deadline
 - unlike downloads, the data may not be consumed at the server right away
 - if a piece of data arrives after the deadline, we just need to guarantee that it's exactly the same piece of data that was committed before the deadline



Copyright © 2008, Spring 2008

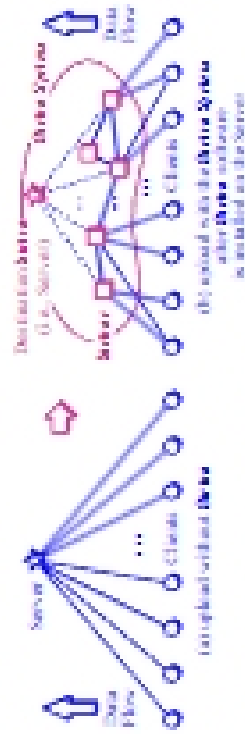
Solution with **Bistro**

- In terms of deadline
 - Client 11 is happy
 - Client 12 is not happy
 - Client 14 is not happy
- Traffic at linear Data-Move-Server
 - Traffic is not a problem
 - Data
 - Deadline



Copyright © 2008, Spring 2008

A Solution to Upload with Deadlines



- A **Bistro** is like an **upload-only** **CDN** built to handle deadline-sensitive uploads
- A **Bistro** can be installed on an **RD** server or a **low** partner's server
 - **local**
 - **Multiple** servers may be going on concurrently, or overlapping, each with a different deadline on server



Copyright © 2008, Spring 2008