

The transcoding proxy

A demonstration

Björn Knutsson
University of Pennsylvania

In cooperation with Honghui Lu, UPenn, and Jeffrey Mogul, HP Western Research Lab

Transcoding basics

- **Transcoding can be used to:**
 - **Reduce complexity of server content**
 - **Reduce bandwidth utilization**
 - **Adapt content to variable client capabilities**



Problem with blind transcoding



**JPEG, 600x800
Q=50, 48617 bytes**



**JPEG, 600x800
Q=3, 5830 bytes**

Need semantic hints!



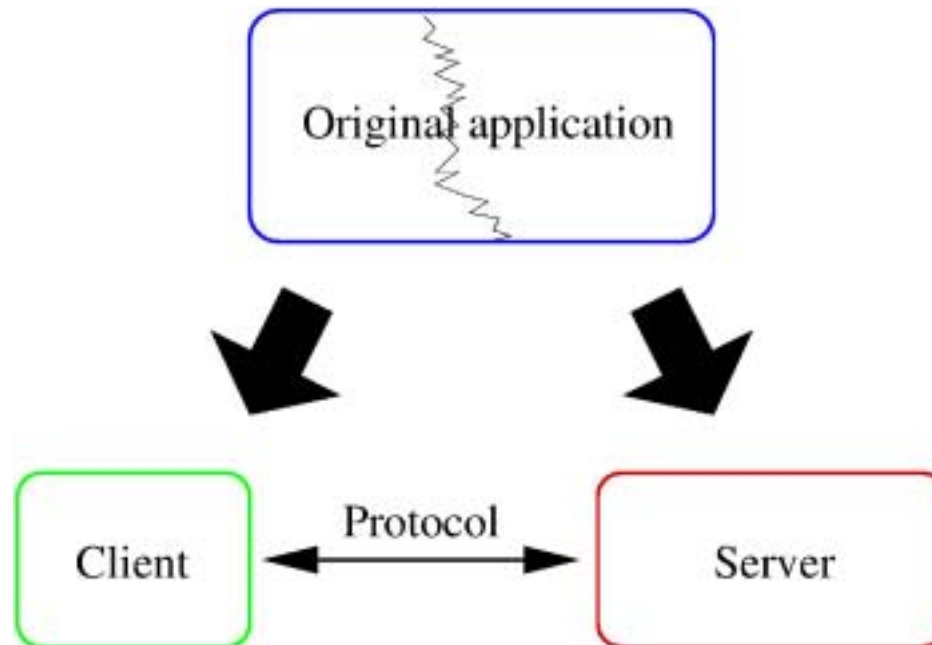
Q=50, 5831 bytes



**JPEG, 200x267
Q=34, 5815 bytes**

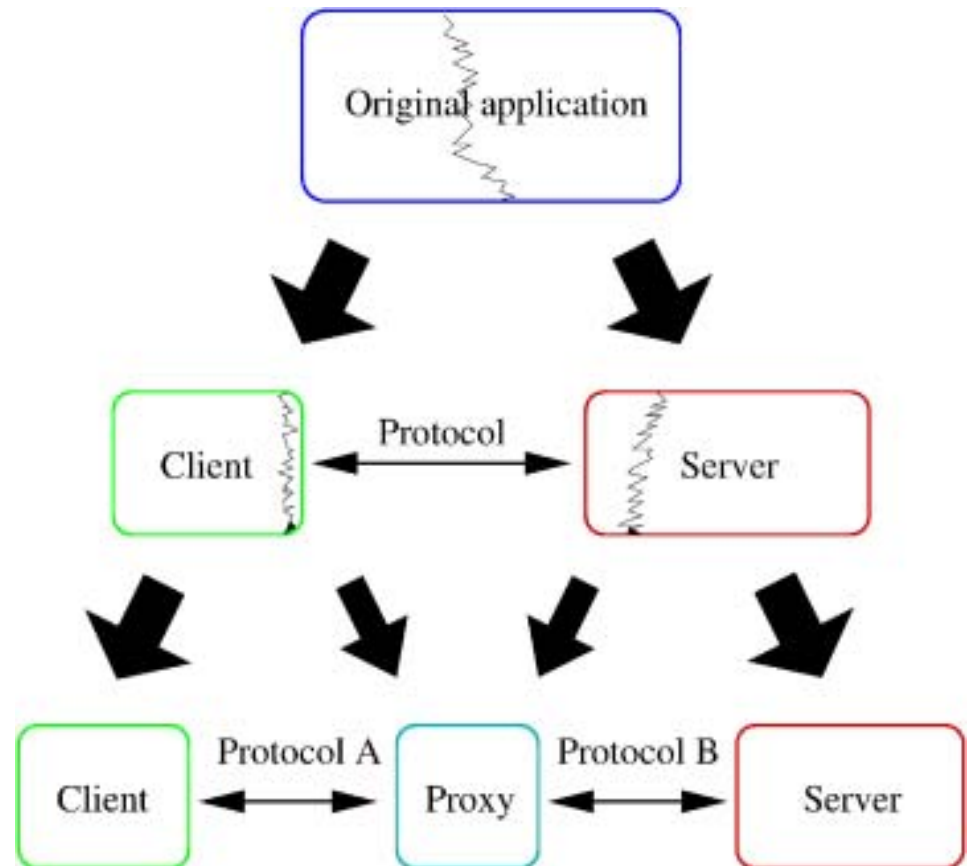
Diffuse computing & distributed systems

- **The client/server model is popular for building distributed systems**
 - **Not very suitable for diffuse computing**
 - **Essentially splits application in two parts with little flexibility in placement or mode of communication**



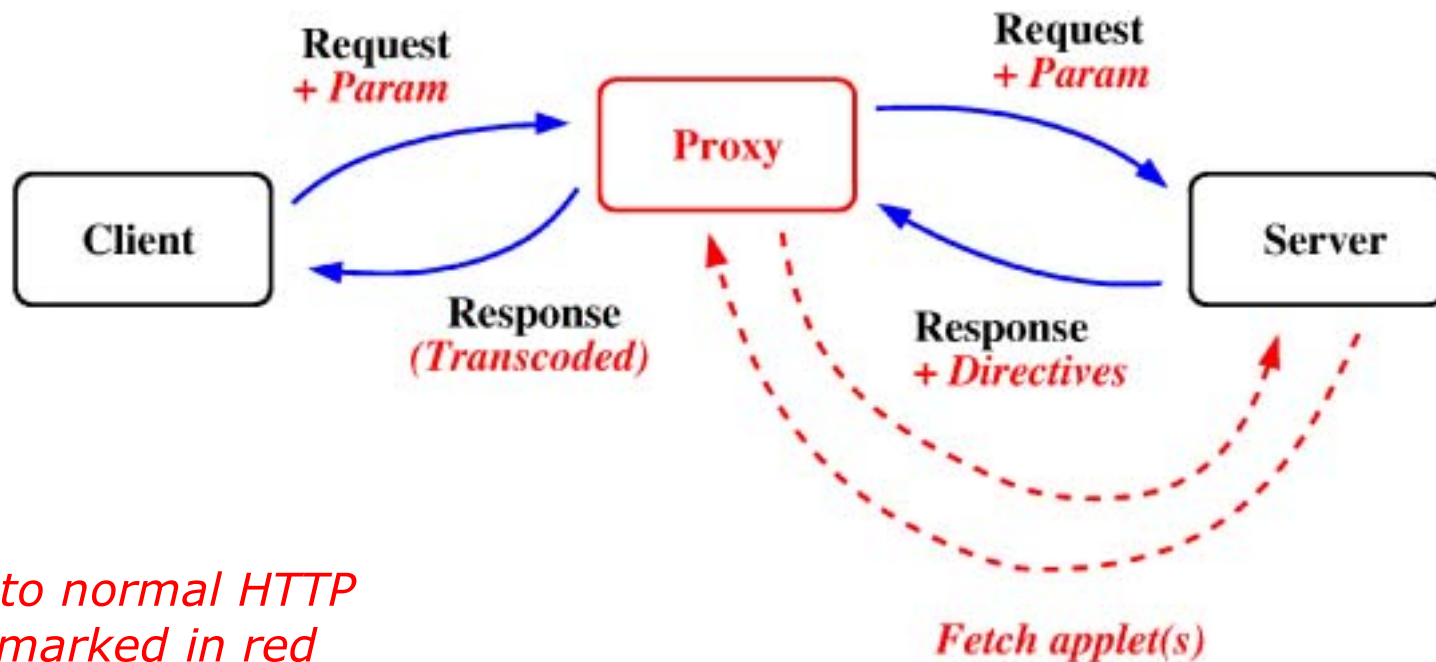
Diffuse computing & distributed systems, cont

- We propose to further divide the application
- The resulting middle part(s) is not *a priori* bound to a specific location
- Decouples client and server, allowing increased flexibility
- Allow us to address scaling issues by shifting processing
- Increased heterogeneity can be compensated by proxies



Transcoding proxy architecture

- Transcoding implemented in proxy between server and client and communicates using extended HTTP
- Transcoding controlled by *self-describing objects*, implemented by encoding *server directives into applets*
- Use of mobile code increases flexibility and decoupling



Additions to normal HTTP protocol marked in red

About the implementations

We have two separate implementations:

- **The Perl-based prototype proxy**
 - Does not cache objects and does not support sandbox
 - Good for testing new applet ideas, can use arbitrary existing commands
 - Have several implemented several applets
- **The Java-based caching proxy**
 - Integrated with the Squid web cache and uses JVM for sandboxing
 - Requires either native implementations or JNI interfaces
 - Still some performance issue (current performance 1/2 of Perl version)

Version demonstrated today is Perl prototype

Let the demos begin...

...and now a short summary.

Summary of demo

- **The demo uses a standard web server (Apache) and standard browsers (Pocket Internet Explorer and Mozilla)**
- **The demo shows that even without caching the proxy is useful and performance acceptable - feasible approach**
- **It is also a demo of our approach to diffuse computing**
 - **Decoupling, scalability, flexibility, heterogeneity**