Optimal File-Distribution in Heterogeneous and Asymmetric Storage Networks

Tobias Langner, Christian Schindelhauer, Alexander Souza

Computer Engineering and Networks Laboratory (TIK)
Department for Information Technology and Electrical Engineering
ETH Zurich, Switzerland

SOFSEM '11
Nový Smokovec

25 January, 2011

Distributed Storage is Ubiquitous

Motivation

- Storing files on multiple machines is a common habit
  - P2P overlays basically represent storage networks
  - File-hosting services like Amazon S3, Rapidshare, etc.

Distributed Storage is Ubiquitous

Motivation

- Storing files on multiple machines is a common habit
  - P2P overlays basically represent storage networks
  - File-hosting services like Amazon S3, Rapidshare, etc.
- Gigantic storage requirements by special applications (Google, Amazon, CERN)

Distributed Storage is Ubiquitous

Motivation

- Storing files on multiple machines is a common habit
  - P2P overlays basically represent storage networks
  - File-hosting services like Amazon S3, Rapidshare, etc.
- Gigantic storage requirements by special applications (Google, Amazon, CERN)
- Most existing approaches specifically tailored for data centers with homogeneous and symmetric network connections.

Distributed Storage is Ubiquitous

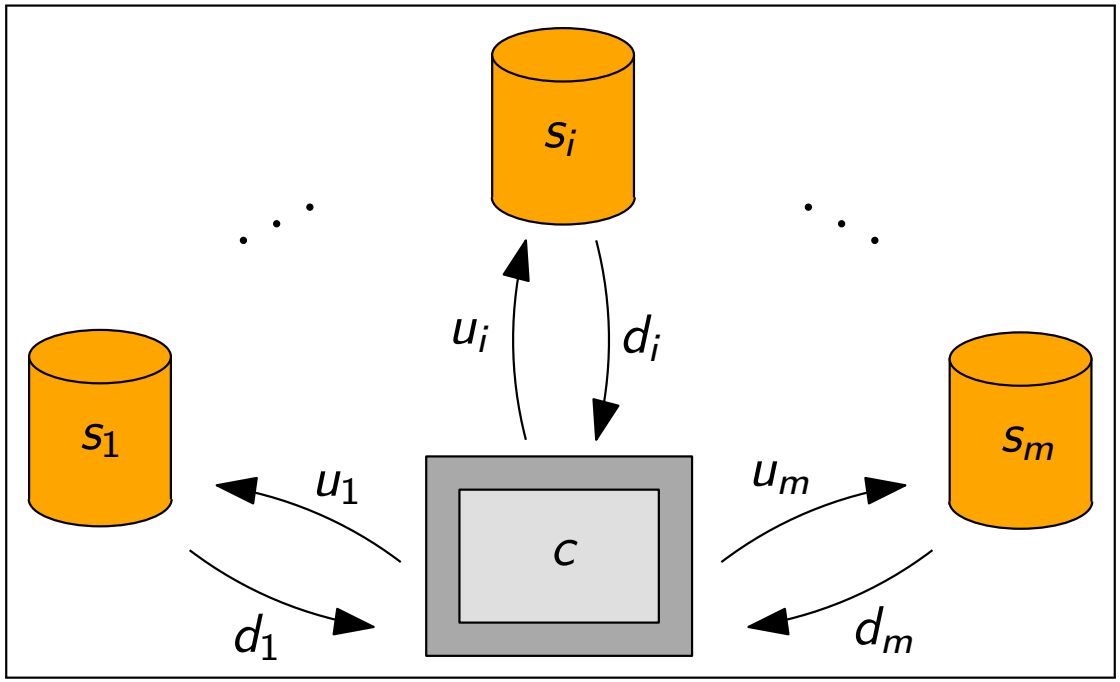
Motivation

- Storing files on multiple machines is a common habit
  - P2P overlays basically represent storage networks
  - File-hosting services like Amazon S3, Rapidshare, etc.
- Gigantic storage requirements by special applications (Google, Amazon, CERN)
- Most existing approaches specifically tailored for data centers with homogeneous and symmetric network connections.
- Little attention to asymmetric bandwidths typical for end-user connections

Synopsis

Distribution Network

Problem Setting

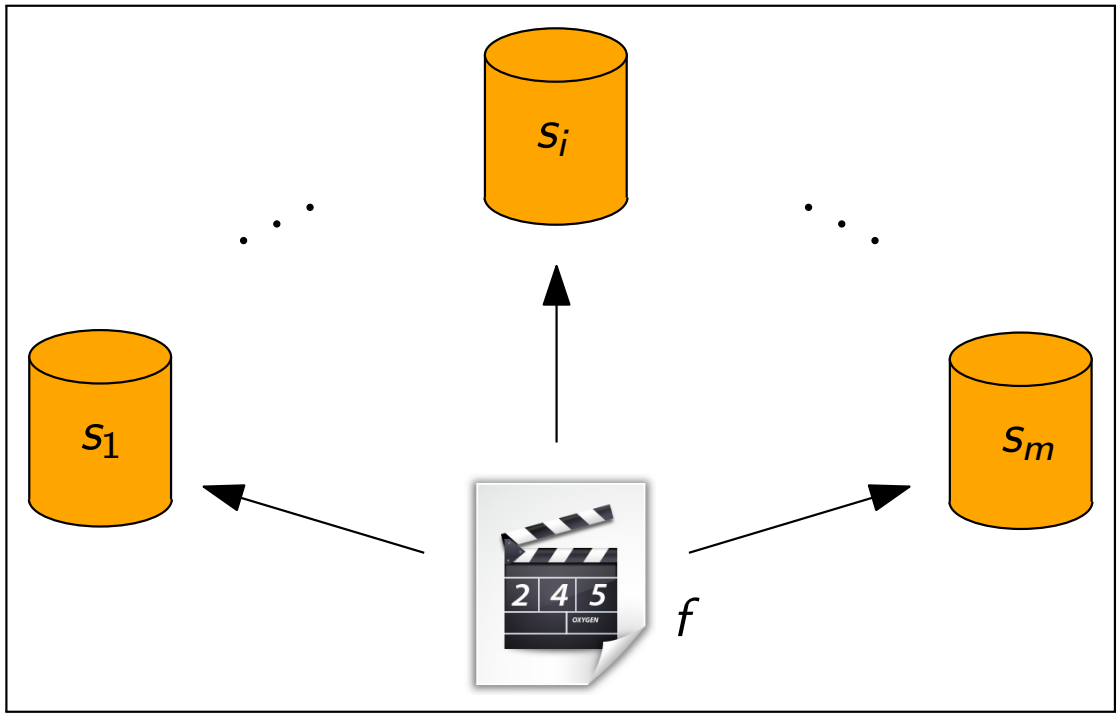


- Servers s_i with $i \in S = \{1, \ldots, m\}$, and one client c
- Each server s_i is characterized by its upload bandwidth u_i and download bandwidth d_i

Distribution Problem

Problem Setting

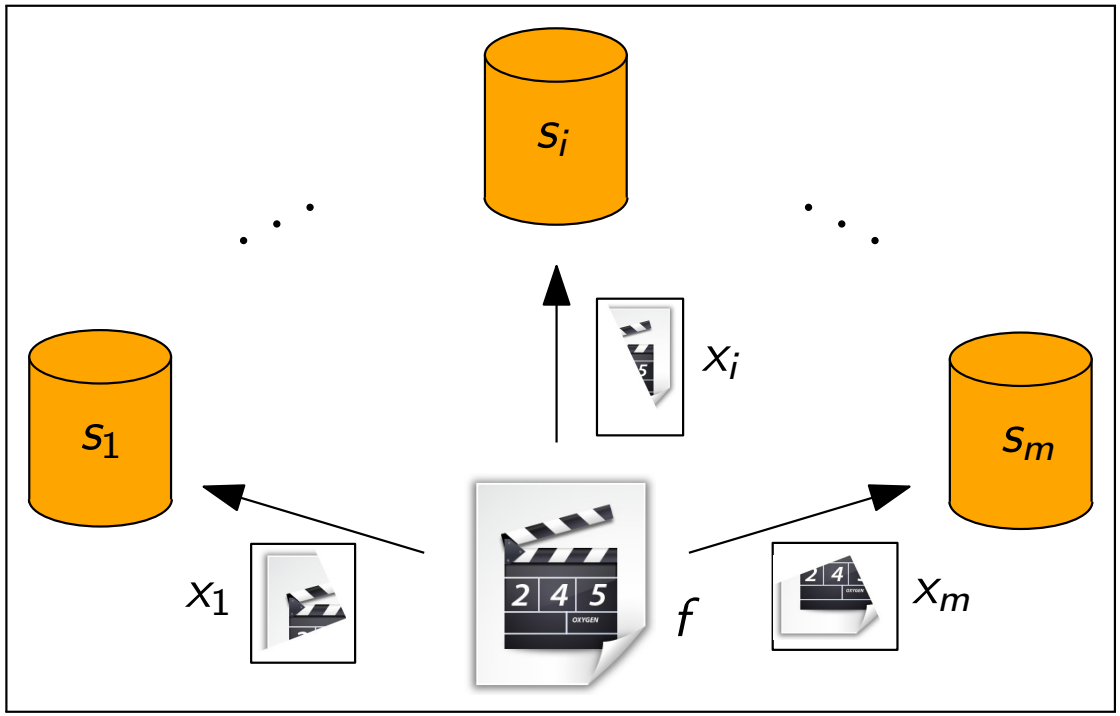
- How do we split a file f of size $|f|$ into fragments for the servers to minimize the time for one upload and n downloads?



Distribution Problem

Problem Setting

- How do we split a file f of size $|f|$ into fragments for the servers to minimize the time for one upload and n downloads?

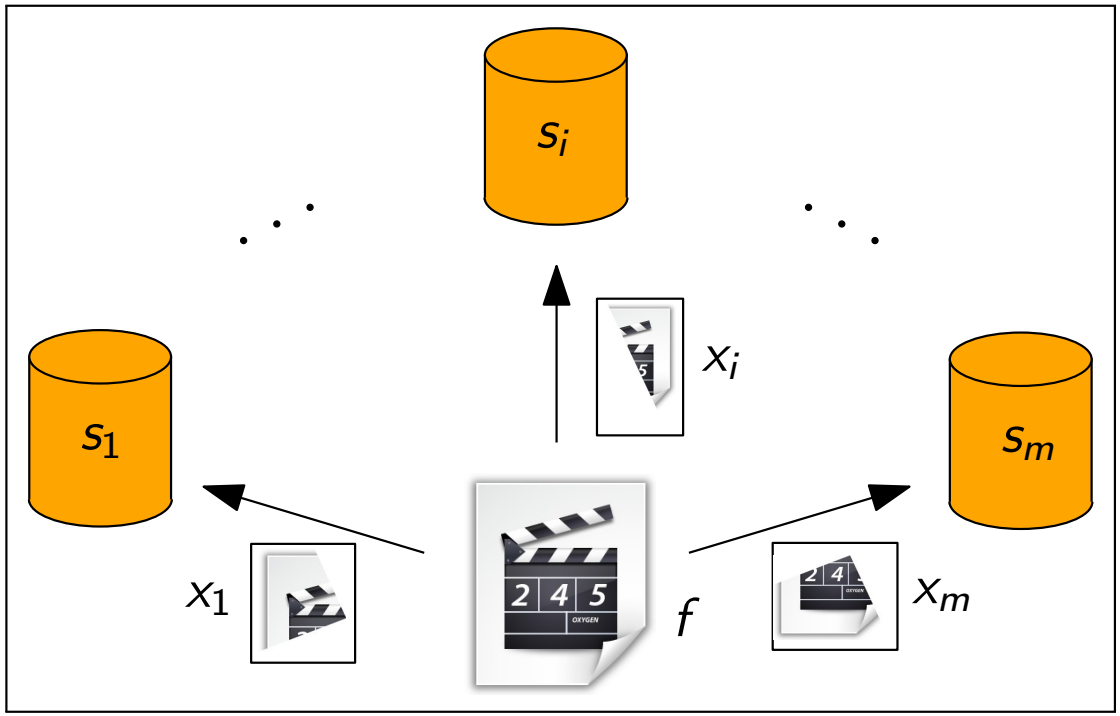


- Distribution vector $\mathbf{x} = (x_1, \ldots, x_m)$ with $\sum x_i = |f|$.
  x_i is the size of the fragment of server s_i.

Distribution Problem

Problem Setting

- How do we split a file f of size $|f|$ into fragments for the servers to minimize the time for one upload and n downloads?



- Distribution vector $\mathbf{x} = (x_1, \ldots, x_m)$ with $\sum x_i = |f|$. x_i is the size of the fragment of server s_i.
- Equivalent simplified problem with $|f| = n = 1$.

Transfer Times

Problem Setting

The quality of a distribution $\mathbf{x}$ is given by the transfer times:

- Upload time:

$$t_u(\mathbf{x}) = \max_{i \in S} \left\{ \frac{x_i}{u_i} \right\}$$

Transfer Times

Problem Setting

The quality of a distribution $\mathbf{x}$ is given by the transfer times:

- Upload time:

$$t_u(\mathbf{x}) = \max_{i \in S} \left\{ \frac{x_i}{u_i} \right\}$$

- Download time:

$$t_d(\mathbf{x}) = \max_{i \in S} \left\{ \frac{x_i}{d_i} \right\}$$

Transfer Times

Problem Setting

The quality of a distribution $\mathbf{x}$ is given by the transfer times:

- Upload time:

$$t_u(\mathbf{x}) = \max_{i \in S} \left\{ \frac{x_i}{u_i} \right\}$$

- Download time:

$$t_d(\mathbf{x}) = \max_{i \in S} \left\{ \frac{x_i}{d_i} \right\}$$

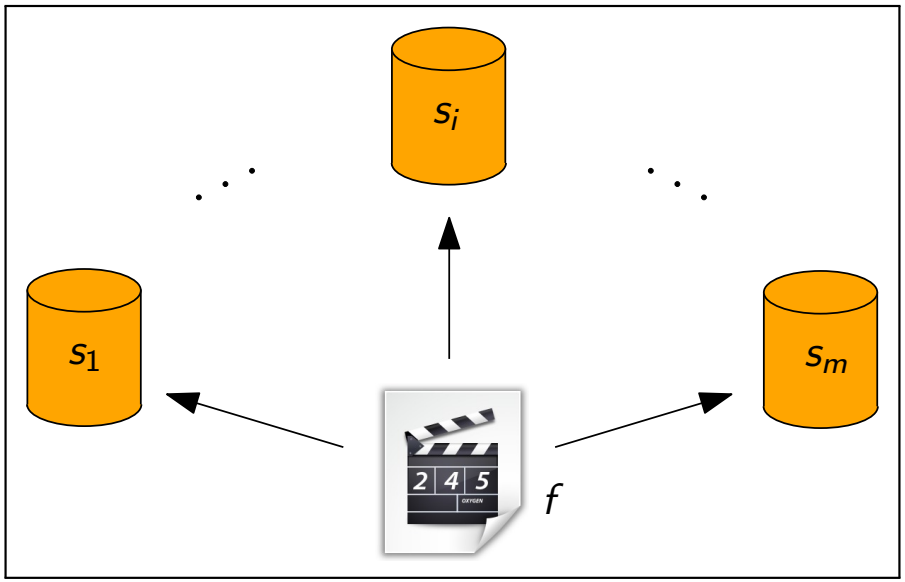
- Total time:

$$\mathrm{val}(\mathbf{x}) = t_u(\mathbf{x}) + (n \cdot)\, t_d(\mathbf{x})$$

Objective

Problem Setting

We are given a distribution network and a file f with size 1.



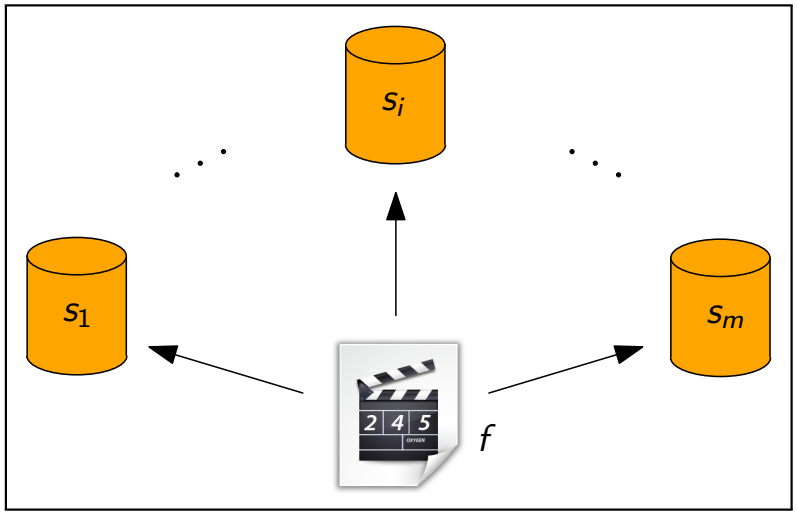
Objective

Find the optimal distribution $\mathbf{x}^*$ that minimizes the total time

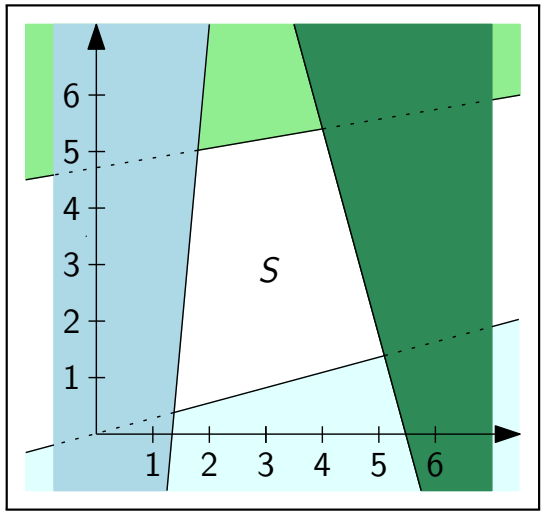
$$\text{val}(\mathbf{x}) = t_u(\mathbf{x}) + t_d(\mathbf{x}) \ .$$

Important Properties

Problem Setting



$\Rightarrow$



- The distribution problem can be formulated as linear program.
  $\Rightarrow$ Its solution space is a convex region.
  $\Rightarrow$ Every local minimum is global.

Non-Linear Optimization Problem

Linear Program

Upload/download bandwidths u_1 to u_m and d_1 to d_m

File distribution $x = (x_1, \ldots, x_m)$

Non-Linear Program

$$\text{minimize} \quad \max\left\{\frac{x_1}{u_1}, \frac{x_2}{u_2}, \ldots, \frac{x_m}{u_m}\right\} + \max\left\{\frac{x_1}{d_1}, \frac{x_2}{d_2}, \ldots, \frac{x_m}{d_m}\right\}$$

$$\text{subject to} \quad \sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \quad \text{for all } i \in \{1, \ldots, m\}$$

Linear Optimization Problem

Linear Program

Linear Program

$$
\begin{aligned}
\text{minimize} \quad & t_u + t_d \\
\text{subject to} \quad & \sum_{i=1}^{m} x_i = 1 \\
& \frac{x_i}{u_i} \leq t_u \qquad \text{for all } i \in \{1, \ldots, m\} \\
& \frac{x_i}{d_i} \leq t_d \qquad \text{for all } i \in \{1, \ldots, m\} \\
& x_i \geq 0 \qquad \text{for all } i \in \{1, \ldots, m\}
\end{aligned}
$$

Dimension: $m + 2$

Synopsis

The Distribution Problem as Flow Problem

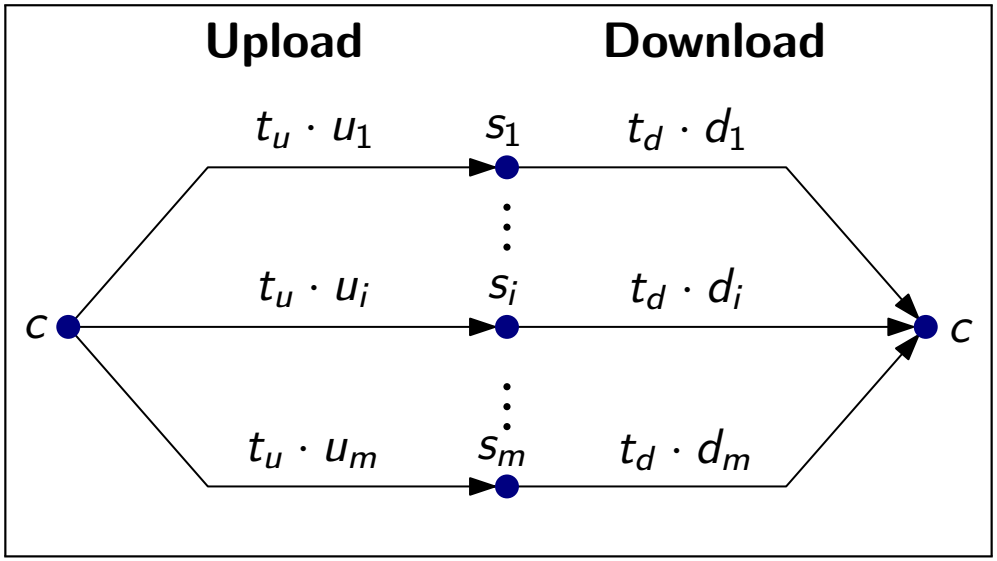
Maximum Flow in Distribution Problems

- Assume we are given upload time t_u and download time t_d

The Distribution Problem as Flow Problem

Maximum Flow in Distribution Problems

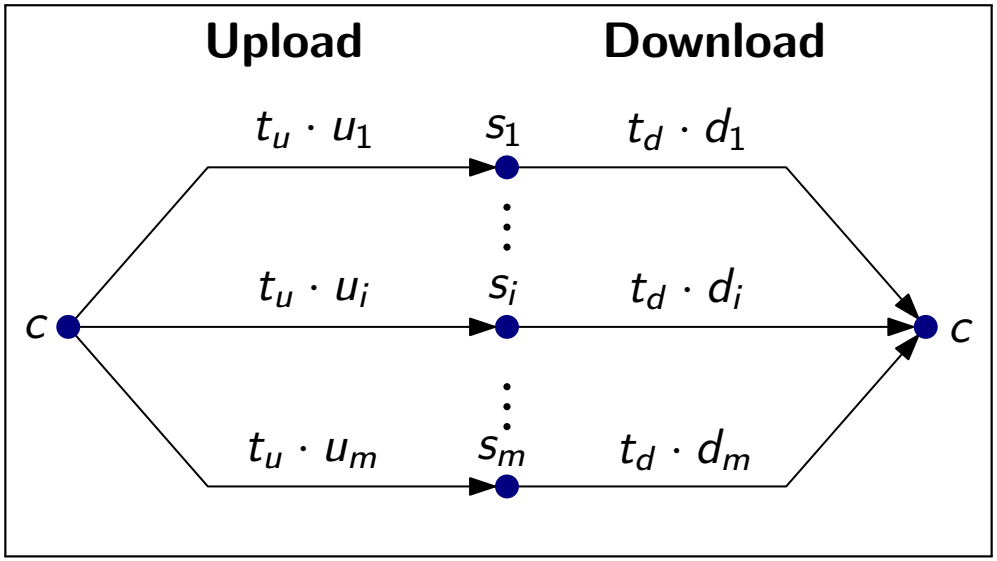
- Assume we are given upload time t_u and download time t_d



The Distribution Problem as Flow Problem

Maximum Flow in Distribution Problems

- Assume we are given upload time t_u and download time t_d

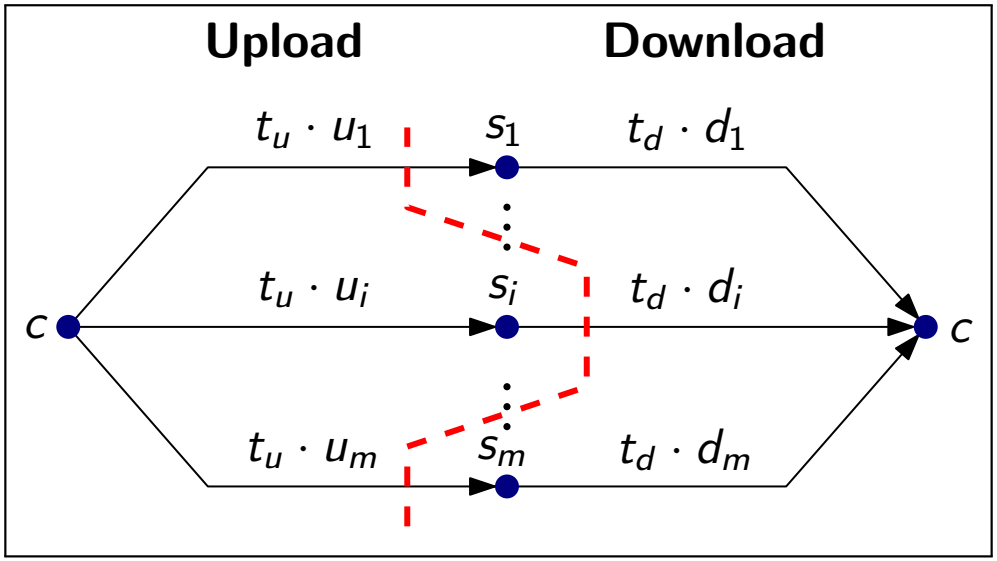


- The maximal amount of data that can be uploaded within t_u and downloaded within t_d corresponds to a maximum flow.

The Distribution Problem as Flow Problem

Maximum Flow in Distribution Problems

- Assume we are given upload time t_u and download time t_d

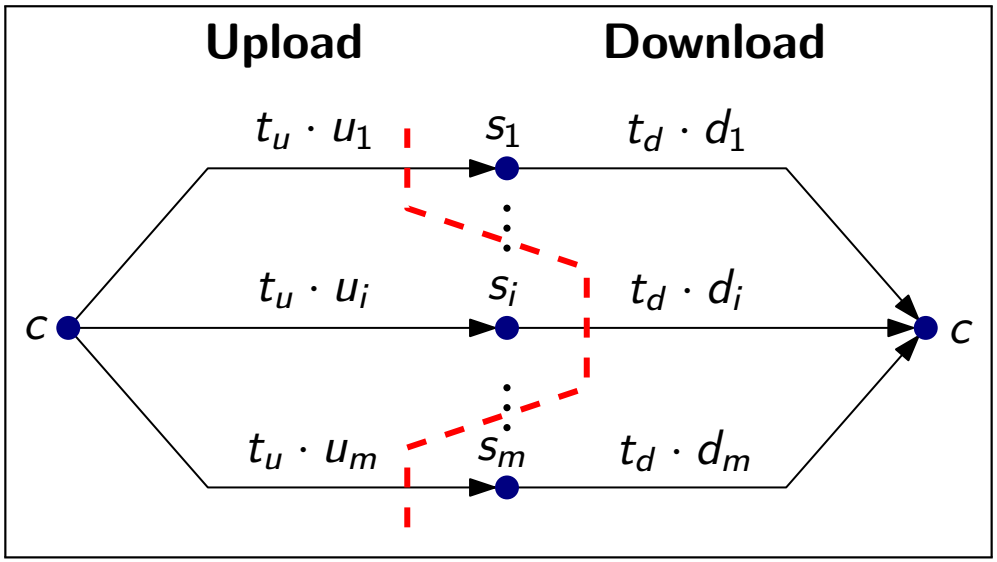


- The maximal amount of data that can be uploaded within t_u and downloaded within t_d corresponds to a maximum flow.
- Value of maximum flow f^* equals capacity of minimal cut

The Distribution Problem as Flow Problem

Maximum Flow in Distribution Problems

- Assume we are given upload time t_u and download time t_d



- The maximal amount of data that can be uploaded within t_u and downloaded within t_d corresponds to a maximum flow.
- Value of maximum flow f^* equals capacity of minimal cut

$$\text{val}(f^*) = \sum_{i=1}^{m} \min\{t_u\, u_i,\ t_d\, d_i\}$$

Decision Predicate

Maximum Flow in Distribution Problems

- There exists a distribution such that f with $|f| = 1$ can be uploaded and downloaded in time t_u and t_d, respectively, if

$$\sum_{i=1}^{m} \min\{t_u\, u_i,\ t_d\, d_i\} \geq 1$$

Decision Predicate

Maximum Flow in Distribution Problems

- There exists a distribution such that f with $|f| = 1$ can be uploaded and downloaded in time t_u and t_d, respectively, if

$$\sum_{i=1}^{m} \min\{t_u\, u_i,\ t_d\, d_i\} \geq 1$$

- We want to find the minimum values for t_u and t_d for which the above predicate holds.

Decision Predicate

Maximum Flow in Distribution Problems

- There exists a distribution such that f with $|f| = 1$ can be uploaded and downloaded in time t_u and t_d, respectively, if

$$\sum_{i=1}^{m} \min\{t_u\, u_i,\ t_d\, d_i\} \geq 1$$

- We want to find the minimum values for t_u and t_d for which the above predicate holds.
- Substitute t_d by $T - t_u$ to obtain the total data function.

$$\delta_T(t_u) = \sum_{i=1}^{m} \min\{t_u\, u_i,\ (T - t_u) \cdot d_i\}$$

Total Data Function

Total Data Function

- The total data function for a fixed value of T:

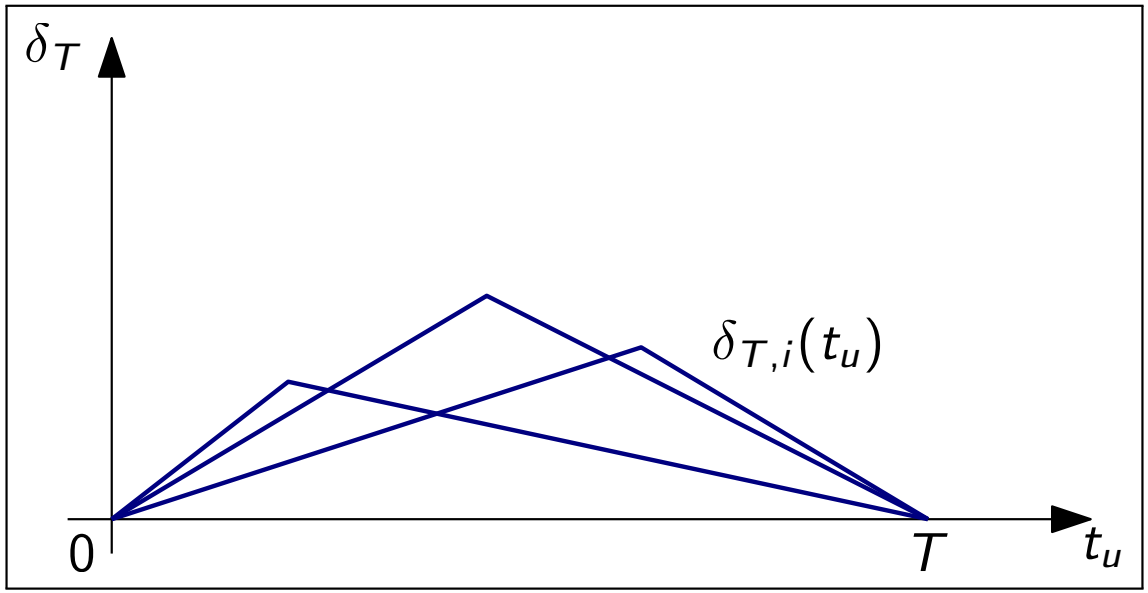
$$\delta_T(t_u) = \sum_{i=1}^{m} \min\{t_u\, u_i,\ (T - t_u) \cdot d_i\}$$

Total Data Function

Total Data Function

- The total data function for a fixed value of T:

$$\delta_T(t_u) = \sum_{i=1}^{m} \min\{t_u\, u_i,\ (T - t_u) \cdot d_i\}$$

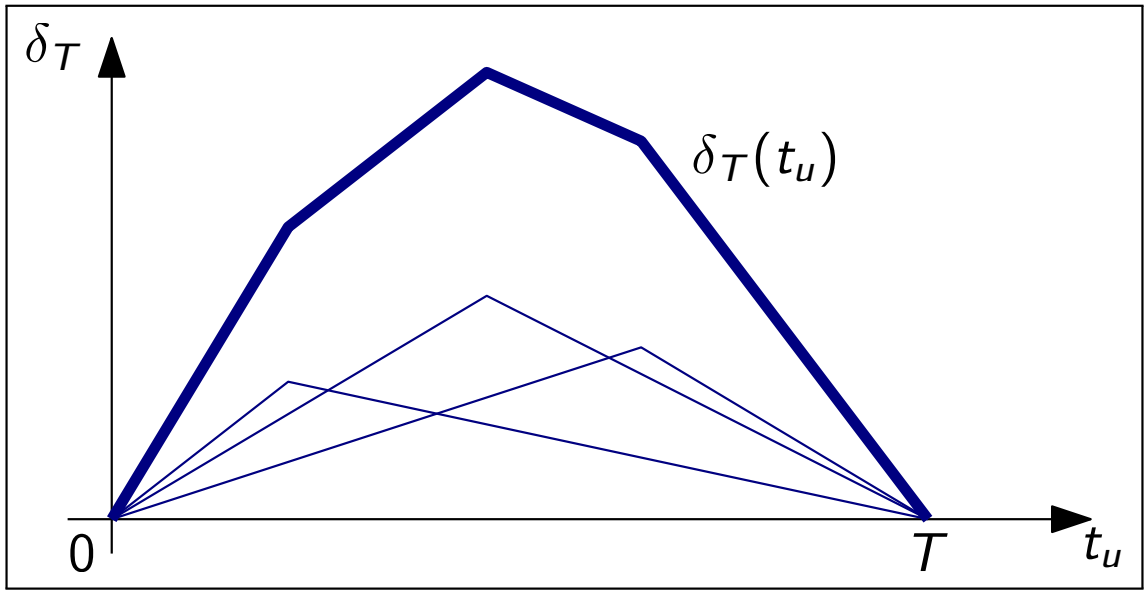


Total Data Function

Total Data Function

- The total data function for a fixed value of T:

$$\delta_T(t_u) = \sum_{i=1}^{m} \min\{t_u\, u_i,\ (T - t_u) \cdot d_i\}$$

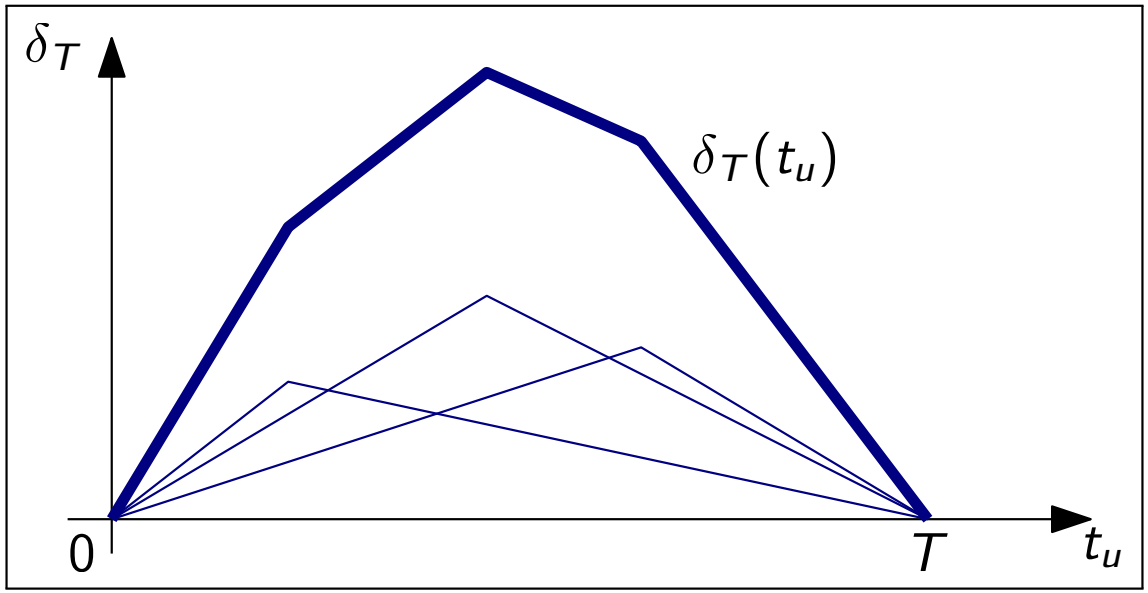


Total Data Function

Total Data Function

- The total data function for a fixed value of T:

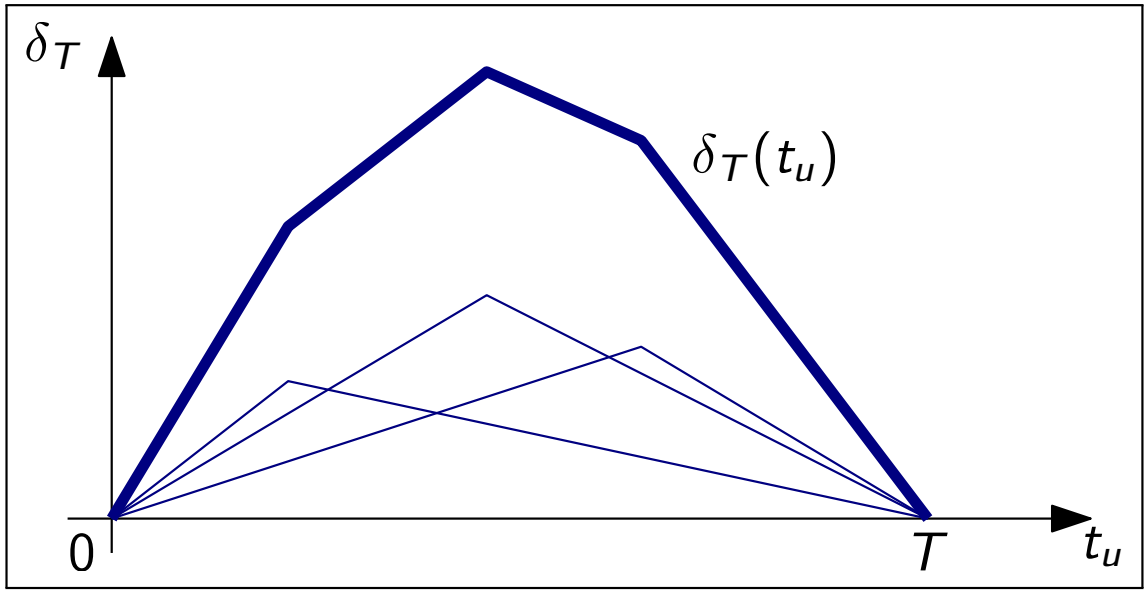
$$\delta_T(t_u) = \sum_{i=1}^{m} \min\{t_u\, u_i,\ (T - t_u) \cdot d_i\}$$



- Can be evaluated in $\mathcal{O}(m \log m)$ steps

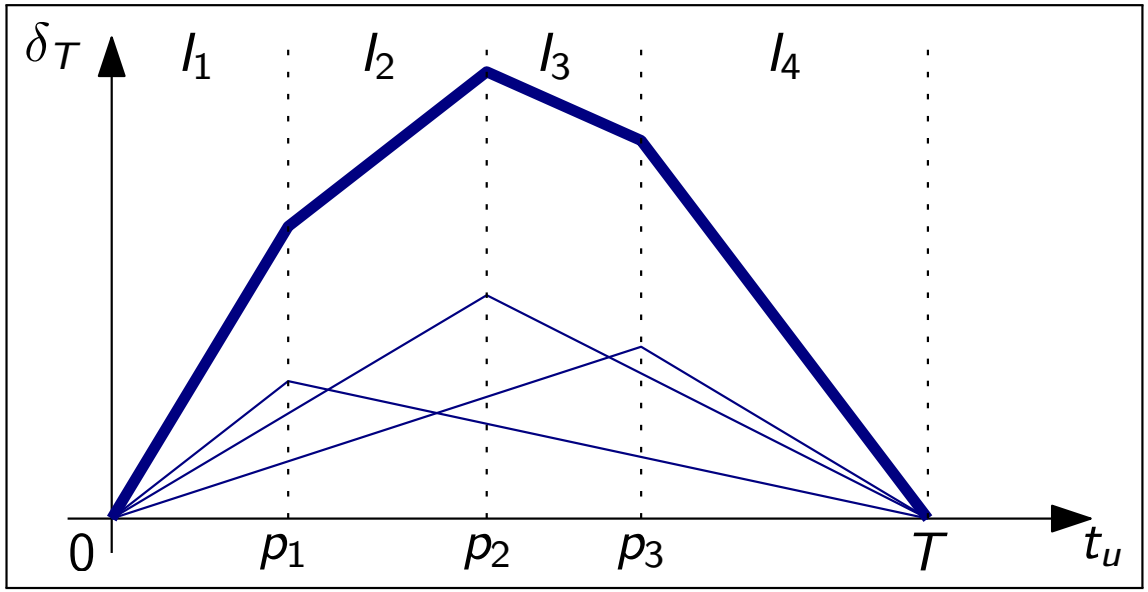
Evaluation

The Total Data Function



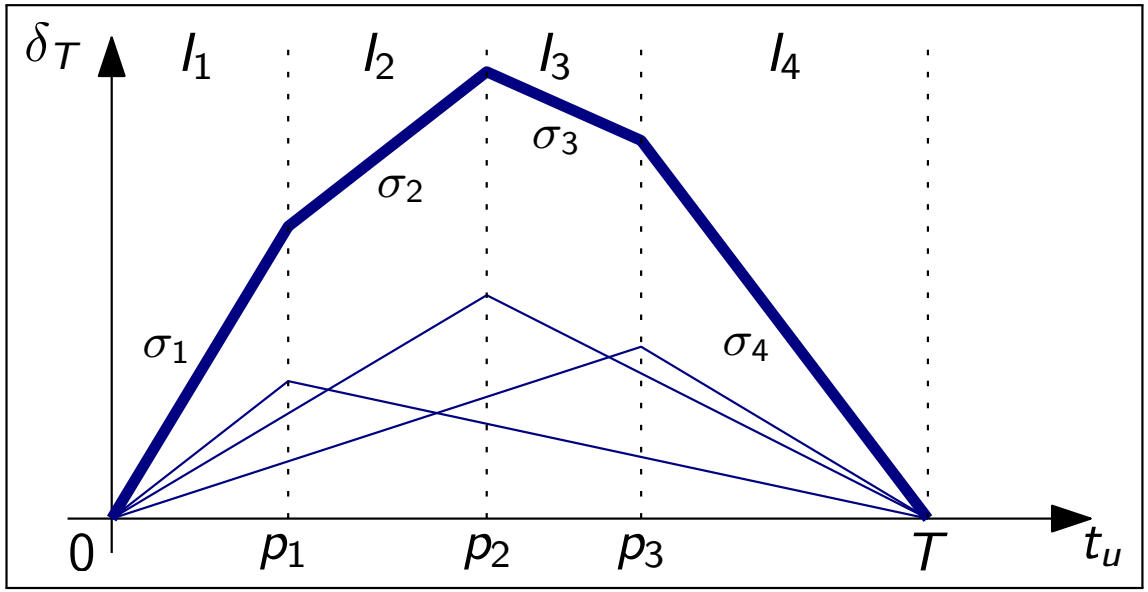
Evaluation

The Total Data Function



Evaluation

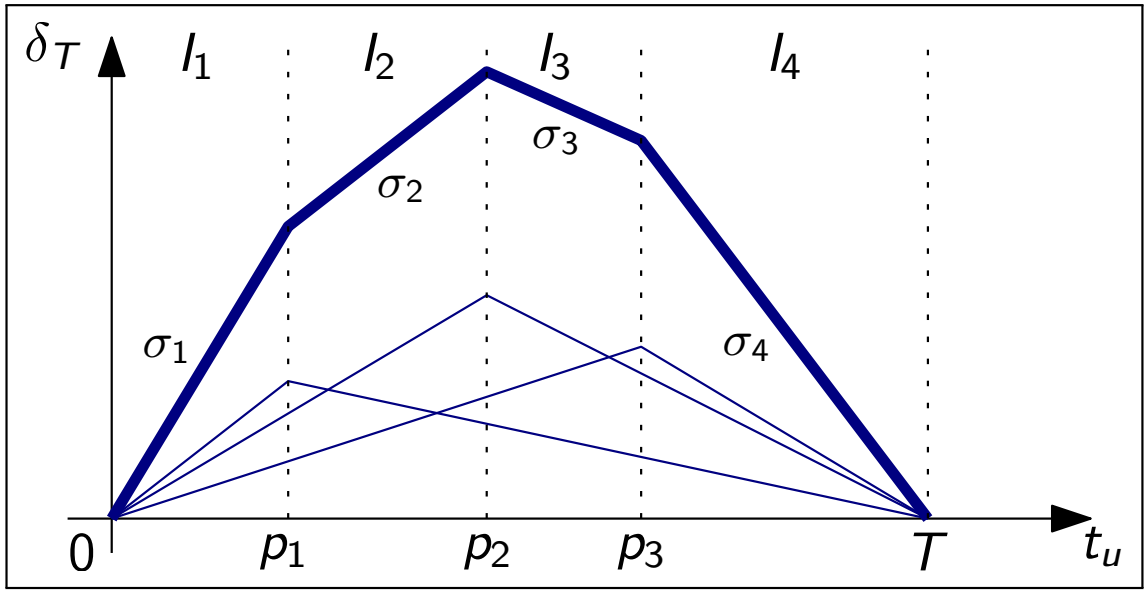
The Total Data Function



σ_i is slope in interval l_i.

Evaluation

The Total Data Function



σ_i is slope in interval l_i.

Recursion formula

$$\delta_T(p_i) = \delta_T(p_{i-1}) + \sigma_i \cdot (p_i - p_{i-1})$$

Scaling the Total Data Function

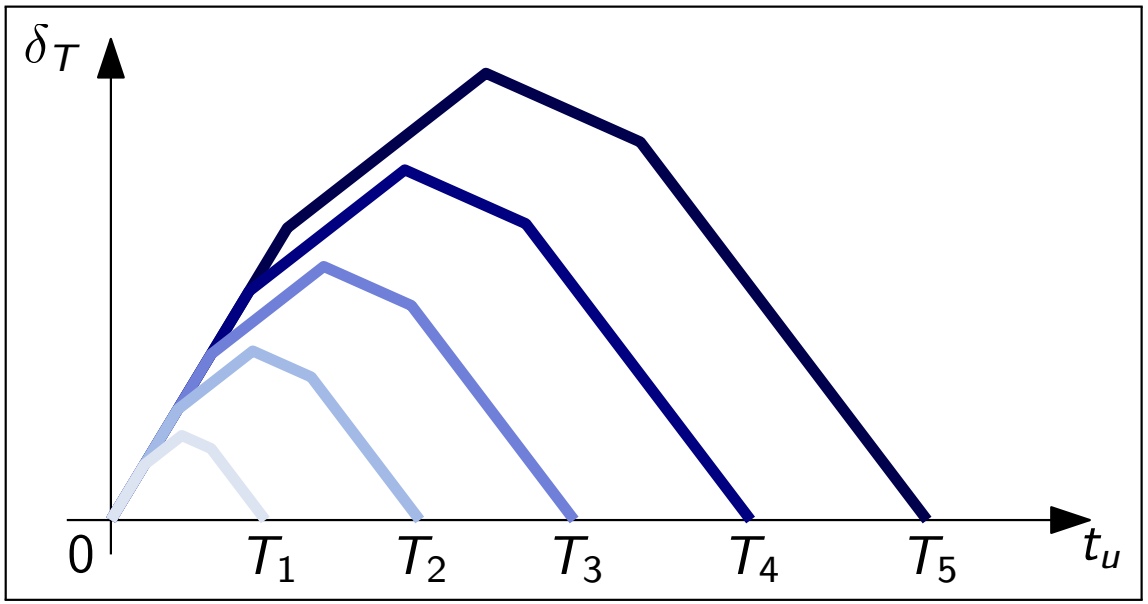
Total Data Function in 3D

- The total data function $\delta_T(t_u)$ determines how much data can be uploaded to and downloaded again from the servers within time T for different values of t_u.

Scaling the Total Data Function

Total Data Function in 3D

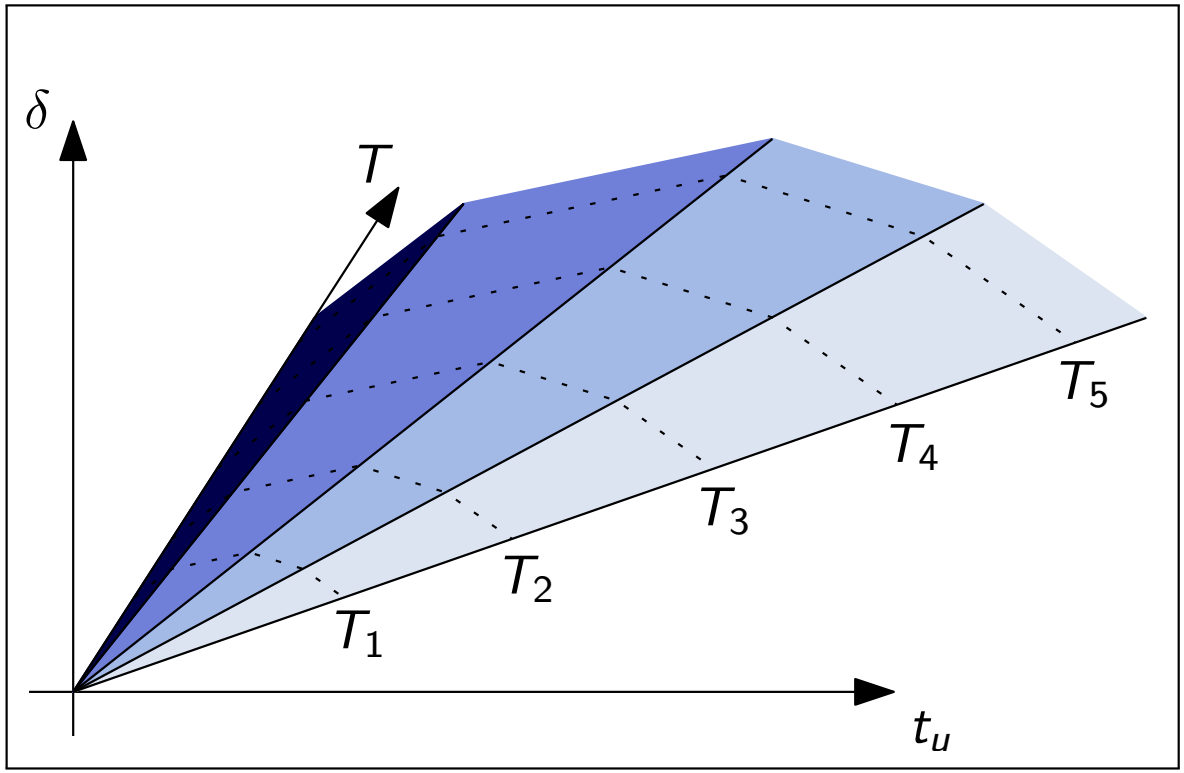
- The total data function $\delta_T(t_u)$ determines how much data can be uploaded to and downloaded again from the servers within time T for different values of t_u.
- Varying values of T only scale the total data function



Two-Variable Total Data Function

Total Data Function in 3D

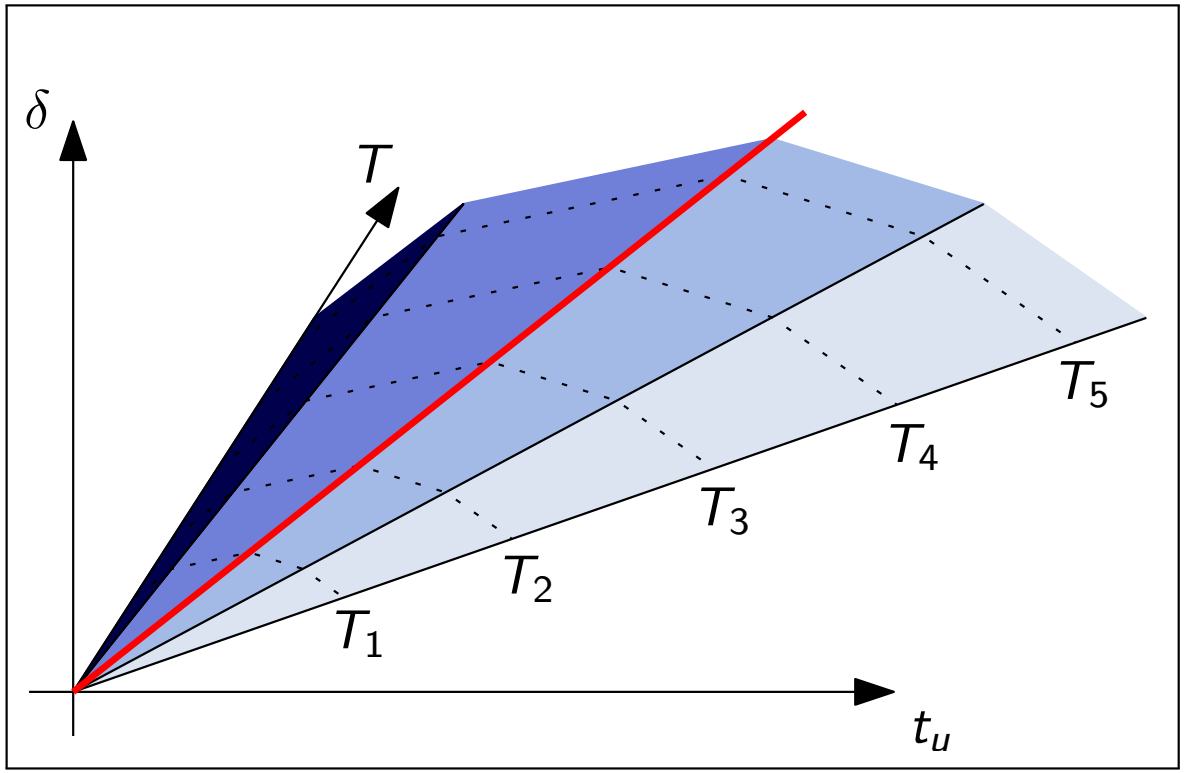
- Interpret $\delta_T(t_u)$ as two-variable function $\delta(T, t_u)$ now



Two-Variable Total Data Function

Total Data Function in 3D

- Interpret $\delta_T(t_u)$ as two-variable function $\delta(T, t_u)$ now

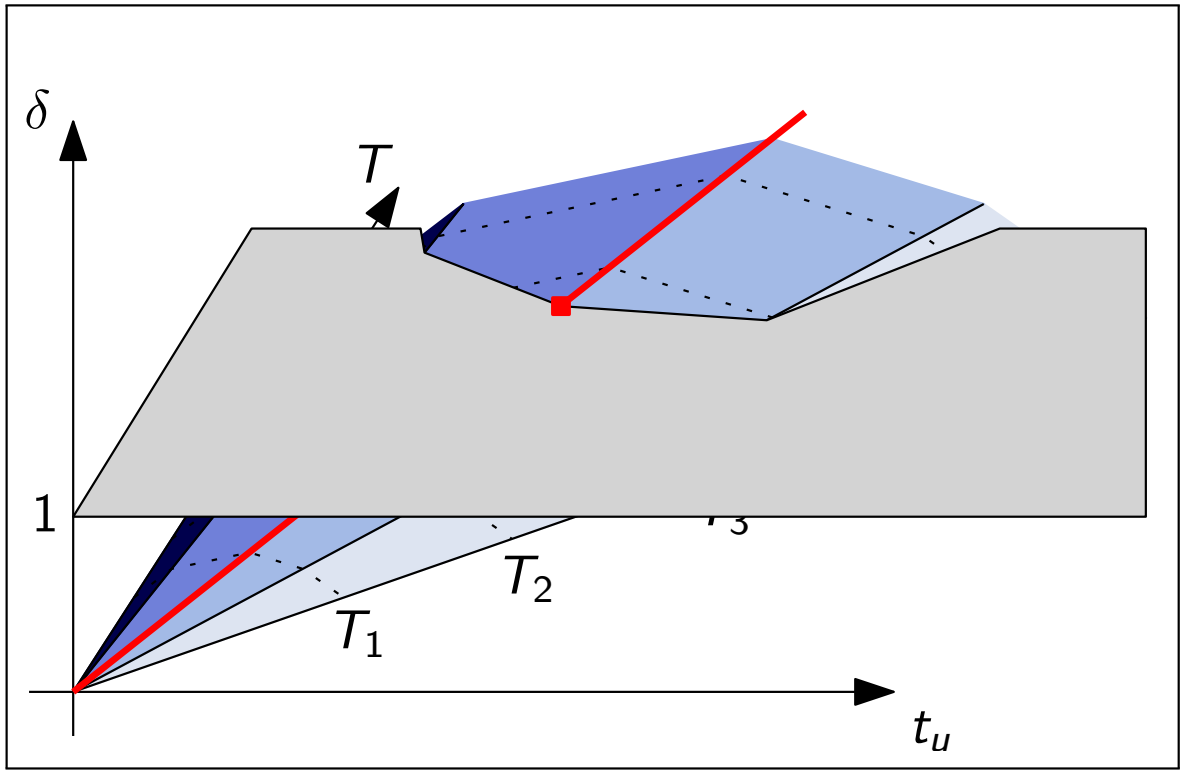


- To find the minimal value of T with $\delta(T, t_u) \geq 1$ for some t_u, establish straight line through the highest point of $\delta(T, t_u)$

Two-Variable Total Data Function

Total Data Function in 3D

- Interpret $\delta_T(t_u)$ as two-variable function $\delta(T, t_u)$ now

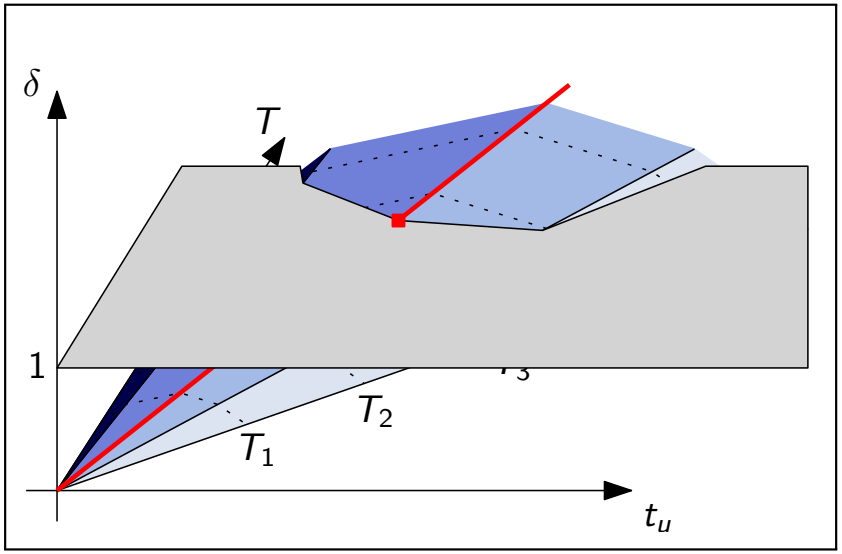


- To find the minimal value of T with $\delta(T, t_u) \geq 1$ for some t_u, establish straight line through the highest point of $\delta(T, t_u)$
- Determine where it intersects the plane $\delta = 1$

Determining the Actual Distribution

The Algorithm

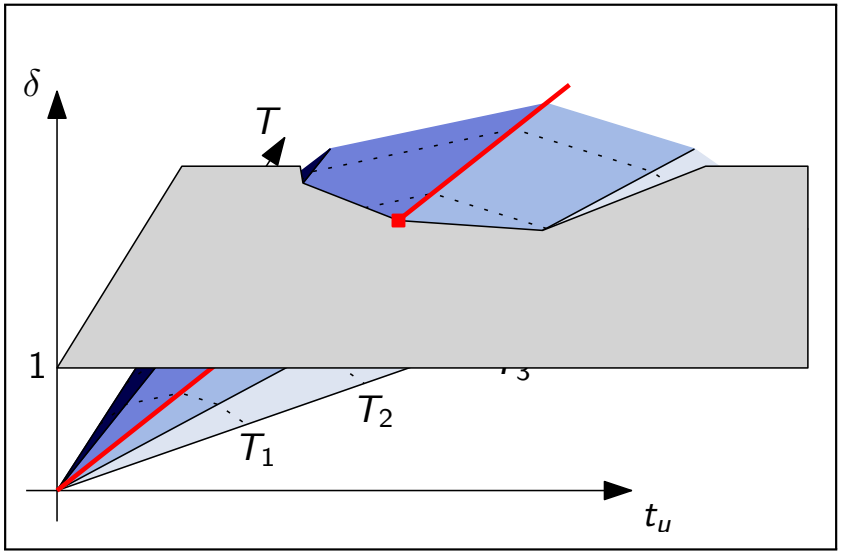
- We determine the minimal value of T as depicted.



Determining the Actual Distribution

The Algorithm

- We determine the minimal value of T as depicted.

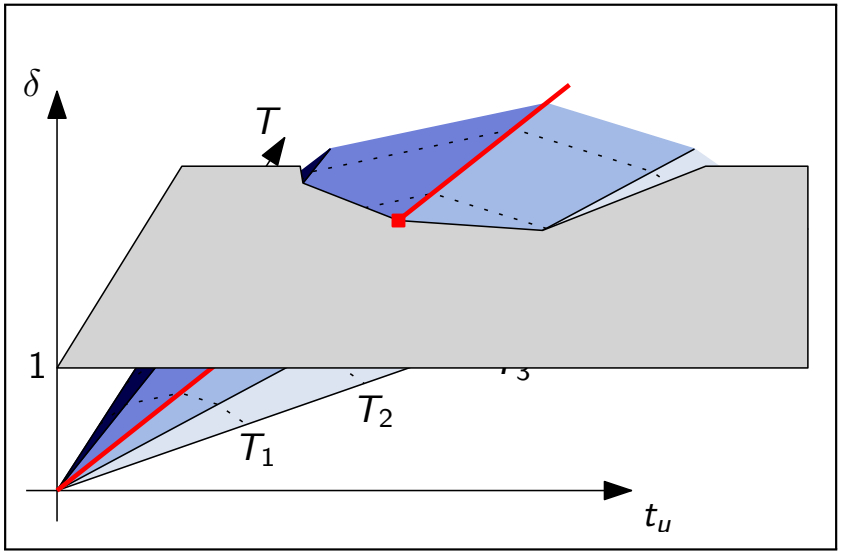


- The decomposition of T into t_u and t_d is given by the coordinates of the intersection point.

Determining the Actual Distribution

The Algorithm

- We determine the minimal value of T as depicted.



- The decomposition of T into t_u and t_d is given by the coordinates of the intersection point.

Determining the Actual Distribution

The Algorithm

- Algorithm to determine a solution with these time bounds:
  - Iterate through all servers $s_i \in S$
  - Assign to s_i the data amount $x_i = \min\{t_u\, u_i,\ t_d\, d_i\}$
  - Return the resulting distribution

Determining the Actual Distribution

The Algorithm

- Algorithm to determine a solution with these time bounds:
  - Iterate through all servers $s_i \in S$
  - Assign to s_i the data amount $x_i = \min\{t_u\, u_i,\ t_d\, d_i\}$
  - Return the resulting distribution
- Through the feasibility predicate

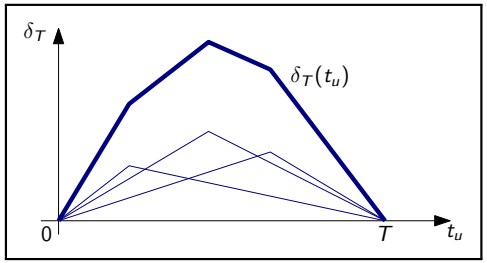
$$\sum_{i=1}^{m} \min\{t_u\, u_i,\ t_d\, d_i\} = 1$$

we know that $\sum x_i = 1$ and thus $\mathbf{x}$ is a valid distribution.

Runtime Complexity

The Algorithm

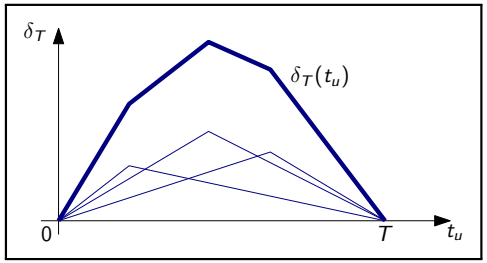
- Evaluating the total data function for a fixed value of T runs in $\mathcal{O}(m \log m)$ steps.



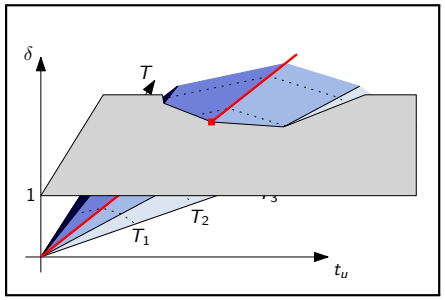
Runtime Complexity

The Algorithm

- Evaluating the total data function for a fixed value of T runs in $\mathcal{O}(m \log m)$ steps.



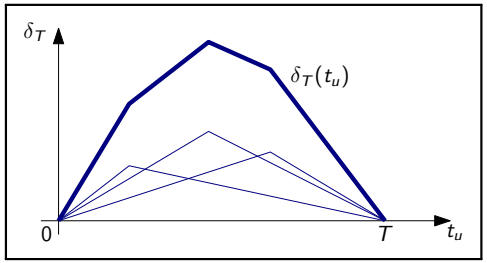
- Intersecting the straight line with the plane $\delta = 1$ is possible in constant time.



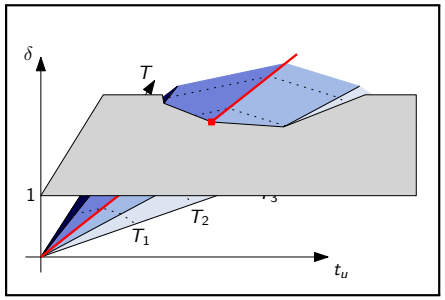
Runtime Complexity

The Algorithm

- Evaluating the total data function for a fixed value of T runs in $\mathcal{O}(m \log m)$ steps.



- Intersecting the straight line with the plane $\delta = 1$ is possible in constant time.



Runtime complexity: $\mathcal{O}(m \log m)$

Synopsis

Summary

- Formal introduction of a new distribution problem

Summary

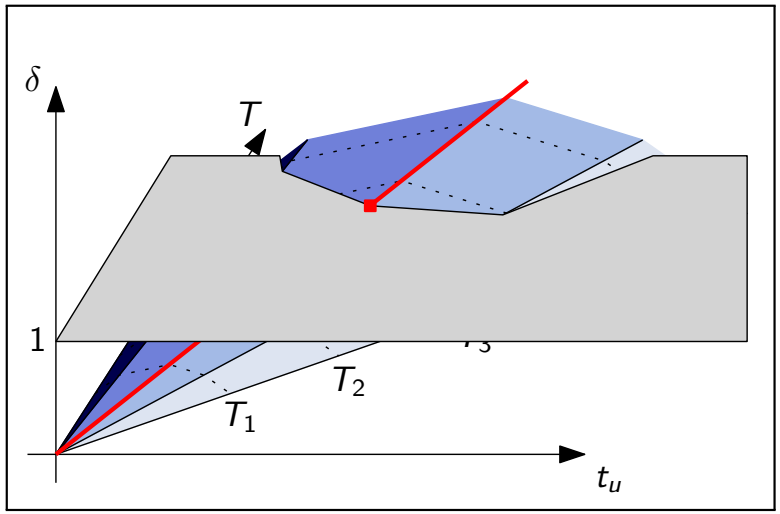
- Formal introduction of a new distribution problem
- Formulation as linear program

Summary

- Formal introduction of a new distribution problem
- Formulation as linear program
- Derived total data function from flow formulation

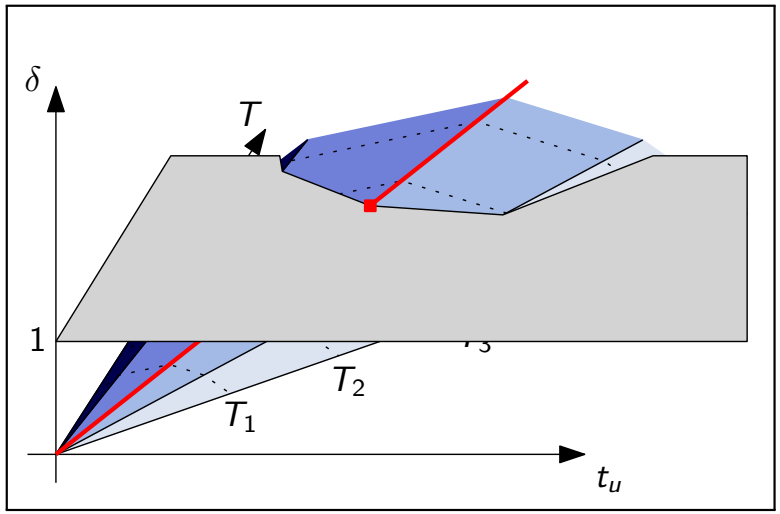
Summary

- Formal introduction of a new distribution problem
- Formulation as linear program
- Derived total data function from flow formulation
- Presented Analytical Scaling algorithm to find optimal solutions

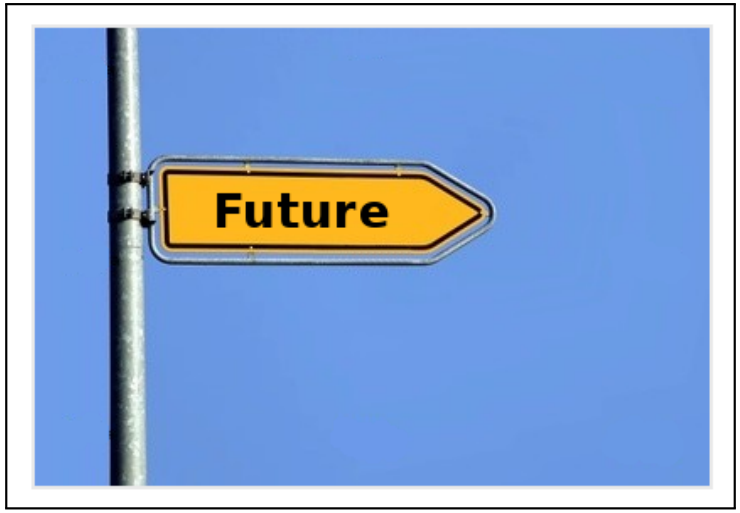


Summary

- Formal introduction of a new distribution problem
- Formulation as linear program
- Derived total data function from flow formulation
- Presented Analytical Scaling algorithm to find optimal solutions

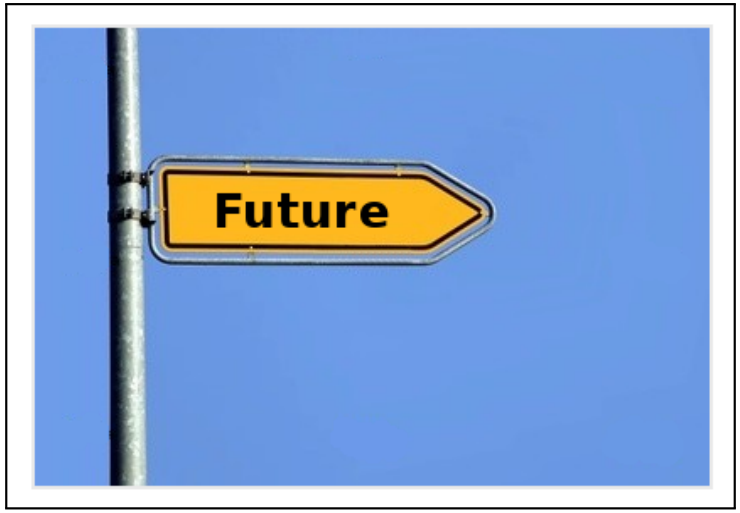


Future Directions



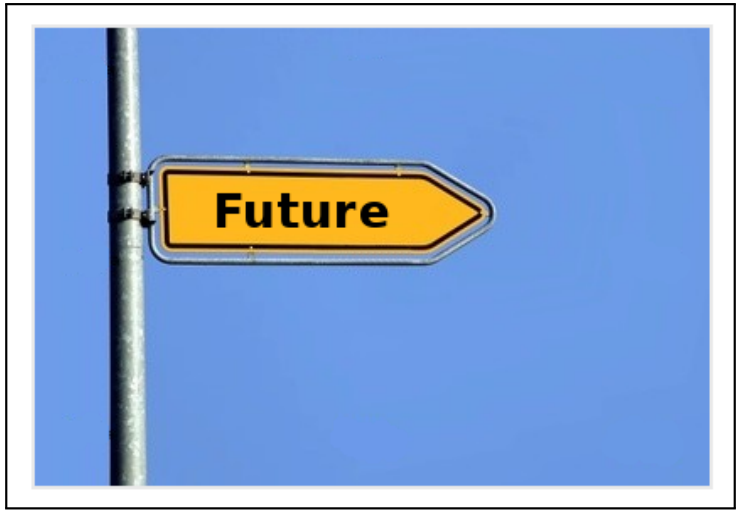
- Implementation of our ideas into real-world applications

Future Directions



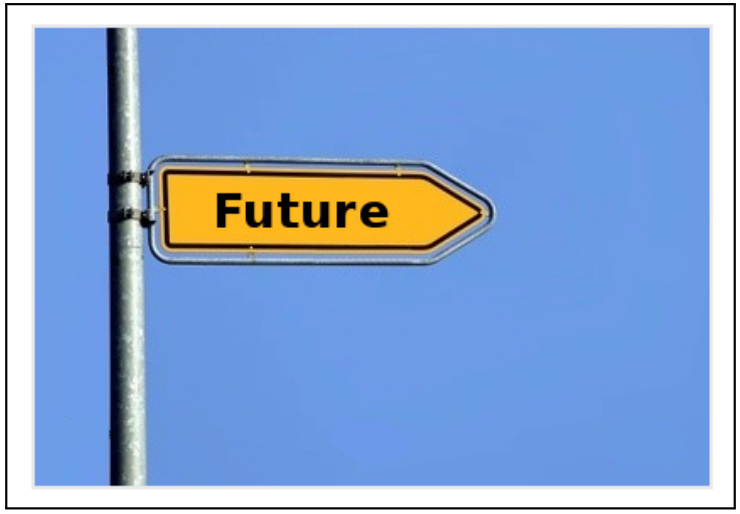
- Implementation of our ideas into real-world applications
- Cope with non-static, fluctuating bandwidths

Future Directions



- Implementation of our ideas into real-world applications
- Cope with non-static, fluctuating bandwidths
- Respect limited storage capacity of servers

Future Directions



- Implementation of our ideas into real-world applications
- Cope with non-static, fluctuating bandwidths
- Respect limited storage capacity of servers
- Incorporation of redundancy to allow for failure tolerance

The End

Thank you for your attention!

Questions?