Performance Evaluation and Asymptotics for Content Delivery Networks

Virag Shah, Prof. Gustavo de Veciana

The University of Texas at Austin

May 02, 2014

Disciplined Engineering of Large Scale CDNs

- ‘Content is King’ [Bill Gates ’96]
- Netflix + Youtube: ~50% of today’s peak internet traffic
  - More than a billion hours of video per month
- Akamai: ~150,000 servers distributed over 1,200 ISPs
  - Delivers 15-30% of all Web traffic, reaching up to 15 TB/s
- Providing better user-perceived performance (download time) at low operating cost is a key problem
- Our Goal: Provide robust models to enable large scale system design, and performance analysis + optimization

Selected Related Work

Large Scale Performance Modeling applicable to CDNs:

- Is routing request to the/a least loaded server enough? [Vvedenskaya et al.'96; Mitzenmacher '96; Bramson et al.'12]

- If we defer service decision until servers become available, how should the number of copies scale? [Tsitsiklis & Xu '13]

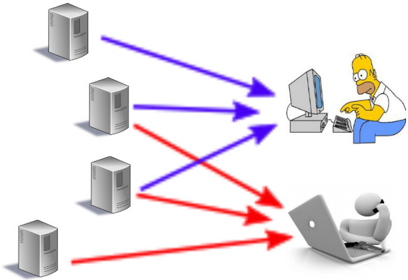
- How should content be replicated/dynamically cached to reduce traffic to centralized back-up? [Leconte et al.'14, Moharir et al.'14]

Other metrics:

- Reliability, e.g., [Cidon et al.'13] show randomized content placement is not always optimal
- Reducing energy costs, e.g., by leveraging energy storage [Palasamudram '12], etc.

The Question

What is the impact on user performance in a large scale system if a subset of servers work together, as a pooled resource, to serve individual download requests?

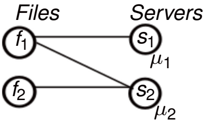


Two key elements:

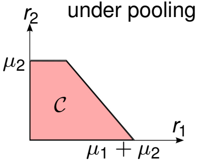
1. Parallel downloads of customer files
2. Coupling across servers

System Model: Simple Example

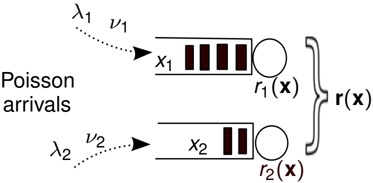
File placement



Capacity region under pooling



Dynamic Model



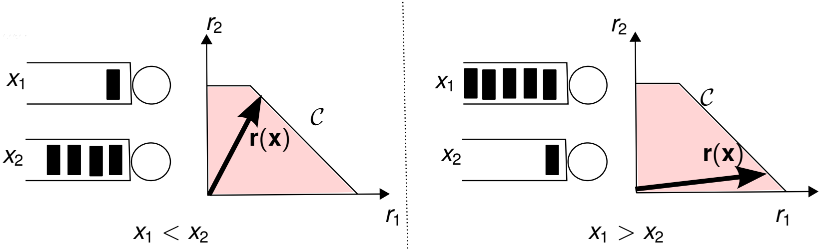
One queue for each file type

Network state $\mathbf{x} = (x_1, x_2)$

Service:

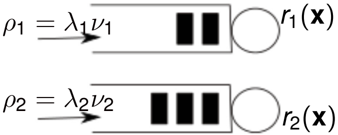
1. rate $r_i(\mathbf{x})$ for i^{th} queue
   – state dependent $\mathbf{r}(\mathbf{x}) \in \mathcal{C} \subset \mathbb{R}_+^2$ for each state $\mathbf{x}$
2. PS discipline within a queue
3. Service requirements: i.i.d. with mean ν_i for file i

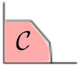
Dynamic Resource Allocation & Fairness



- Ideally, $\mathbf{r}(\mathbf{x})$ assigns more rate to bigger queues, i.e., to file types with more requests
- e.g., Max-min, Proportional, α-fair, Balanced fair
- We use **Balanced fair** because it is close to Proportional fair & tractable
  [e.g., Bonald & Proutiere '03, Massoulié '07, Joseph & de Veciana '11]

What is Balanced Fairness?





$\mathbf{e}_i :=$ unit vector in i^{th} direction

$$r(\mathbf{x}) = \mu$$

$$\pi(\mathbf{x}) = \rho^x(1-\rho)$$

insensitive to service requirement distribution

For some function $\Phi(.) : \mathbb{Z}_+^2 \to \mathbb{R}_+$,

$$r_i(\mathbf{x}) = \frac{\Phi(\mathbf{x}-\mathbf{e}_i)}{\Phi(\mathbf{x})}$$

$$\pi(\mathbf{x}) = \Phi(\mathbf{x})\rho_1^{x_1}\rho_2^{x_2}\,(G(\rho_1,\rho_2))^{-1}$$

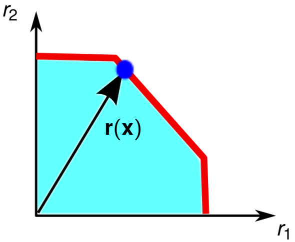
insensitive to service requirement distribution

What is Balanced Fairness?

- Balanced fair rate allocation is the choice for $\Phi(.)$ such that

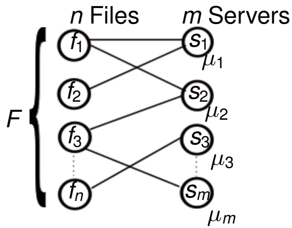
$$\forall \mathbf{x}, \quad \mathbf{r}(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x})) = \left(\frac{\Phi(\mathbf{x} - \mathbf{e}_1)}{\Phi(\mathbf{x})}, \frac{\Phi(\mathbf{x} - \mathbf{e}_2)}{\Phi(\mathbf{x})} \right)$$

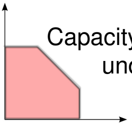
and is on the boundary of $\mathcal{C}$.



- Similarly, generalizes to n dimensions

Content Delivery System Model: General

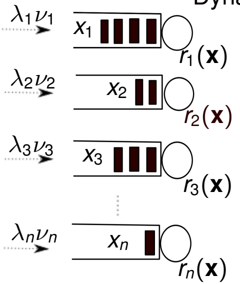




Capacity region $\mathcal{C} \subset \mathbb{R}^n$ under pooling

Dynamic Model

Poisson file requests



$\mathbf{r}(\mathbf{x}) \in \mathcal{C}$ for each state $\mathbf{x} \in \mathbb{Z}^n$ given by balanced fairness

Structural Result: Polymatroid Capacity Region

Theorem

For a given file placement, the capacity region $\mathcal{C}$ is a **polymatroid** *with* **rank function $\mu(.)$.**

- **Rank function:** $\mu : 2^F \to \mathbb{R}_+$, where $\mu(A) :=$ sum capacity for servers that can serve any file in set A
- $\mathcal{C} = \{\mathbf{r} \geq \mathbf{0} : \sum_{i \in A} r_i \leq \mu(A), \ \forall A \subset F\}$
- $\mu(.)$ is submodular, i.e., $\mu(A) + \mu(B) \geq \mu(A \cup B) + \mu(A \cap B)$

Performance Result: Expression for Mean Delay for Serving File Requests

Theorem

Given capacity region $\mathcal{C}$ with rank function $\mu(.)$, and $\boldsymbol{\rho} = (\rho_i : \rho_i = \lambda_i \nu_i, i \in F)$, the mean delay for file f_i is:

$$E[D_i] = \frac{\nu_i \frac{\partial}{\partial \rho_i} G(\boldsymbol{\rho})}{G(\boldsymbol{\rho})}$$

- *where $G(\boldsymbol{\rho}) = \sum_{A \subset F} G_A(\boldsymbol{\rho})$,*
- *and where $G_{\emptyset}(\boldsymbol{\rho}) = 1$, and $G_A(\boldsymbol{\rho})$ can be computed recursively as $G_A(\boldsymbol{\rho}) = \frac{\sum_{i \in A} \rho_i G_{A \setminus \{i\}}(\boldsymbol{\rho})}{\mu(A) - \sum_{j \in A} \rho_j}$.*

Complexity of Expression for Mean Delay

- *Bad news:* $\mu(.)$ has exponential complexity in *n*, so mean delay is hard to compute
- *Good news:* If $\mu(.)$ is symmetric, then complexity is linear in *n*
- *Better news:* Some asymmetric large systems can be asymptotically approximated by that of a symmetric system
    - We now use this idea to analyze practically relevant large scale systems!

Large-Scale CDN Asymptotic Regime and Randomized File Placement

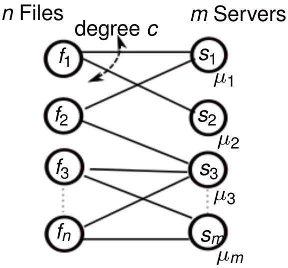
- Large number of server: $m \to \infty$
- Larger number of files: $n \to \infty$ faster than m
- Fixed number of copies for each file: c
  - Stored at random across servers

Load & Capacity: Homogeneity, Scaling and Stability

- Homogeneity of Load and Scaling:
  - Arrival rate for each file: $\lambda^{(m,n)} = \frac{\lambda m}{n}$ for some constant λ
  - Arrival rate per server: λ
  - Mean service requirements for requests of each file: ν
  - Load per server: $\rho = \lambda\nu$, a constant.

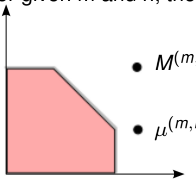
- Homogeneity of Capacity and Stability:
  - Let $\mu_i = \xi$ for each server i
  - Let $\rho < \xi$ for stability.

RPBF Systems: Randomized Placement and Balanced Fairness



Random bipartite graph

For given m and n, the capacity region $\mathcal{C}^{(m,n)}$ is random



- $M^{(m,n)}(.)$:= associated random rank function
- $\mu^{(m,n)}(.)$:= a realization of $M^{(m,n)}(.)$

Given a realization of the Randomized Placement, we study the performance under Balanced Fair rate allocation

Approximation via 'Averaged' Capacity

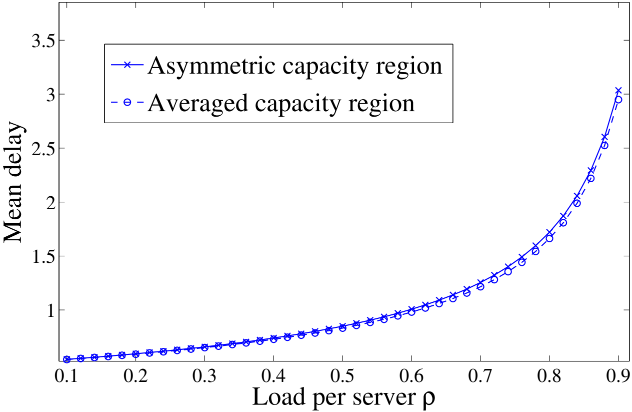
- In a randomized file placement, the averaged rank function

$$\bar{\mu}^{(m,n)}(A) := E[M^{(m,n)}(A)] \text{ for all } A \subset F,$$

is symmetric!

Goodness of Approximation via ‘Averaged’ Capacity

$m = 4$, $n \to \infty$, $c = 2$, and $\xi = 1$



Mean Delay of RPBF Systems: Asymptotics via 'Averaged' Capacity

Theorem

For the 'averaged' capacity region with rank function $\bar{\mu}(.)$, under load homogeneity, scaling and stability assumptions, the expected delay satisfies:

$$\lim_{m\to\infty} \lim_{n\to\infty} E[D^{(m,n)}] = \frac{1}{\lambda c} \log\left(\frac{1}{1-\rho/\xi}\right)$$

- Compare this with standard $M/GI/1$ PS queue where

$$E[D] \propto \frac{1}{1-\rho/\xi}$$

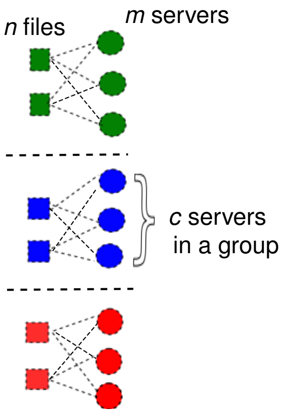
- In $M/GI/1$, total service rate across jobs is fixed
- In RPBF systems, effective service rate increases with more jobs!

Performance Evaluation: Key factors

1. Parallel downloads from servers
   - Abstracted in capacity region $\mathcal{C}^{(m,n)}$
2. Coupling across servers
   - Randomized placement $\implies$ overlapping pools of servers

Claim: BF over $\mathcal{C}^{(m,n)}$ nicely exploits both for load balancing across servers!

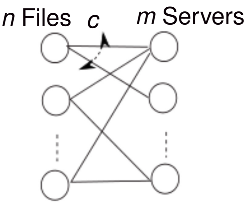
Baseline Policy 1: Fixed Pools and Parallel Downloads



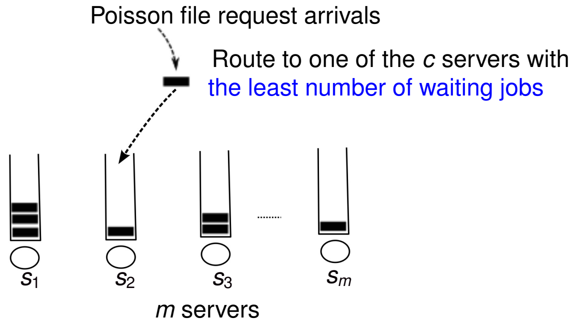
- Pro: Parallel download from servers
- Con: Non-overlapping pools $\Longrightarrow$ No load balancing

Baseline Policy 2: Random Placement and Least-loaded Routing

Randomized File Placement



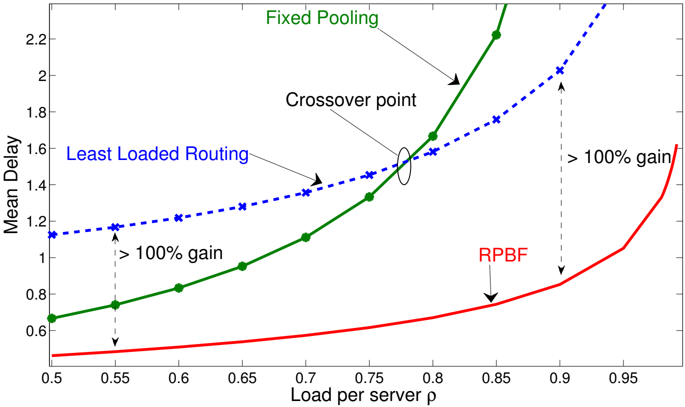
Least-loaded Routing



- Pro: Good load balancing across servers
- Con: No parallel downloads from multiple servers

Performance Comparison

- $c = 3$, $\xi = 1$, $\nu = 1$
- $n \to \infty$ and then $m \to \infty$
- Each policy is stable for $\rho < 1$



Summary

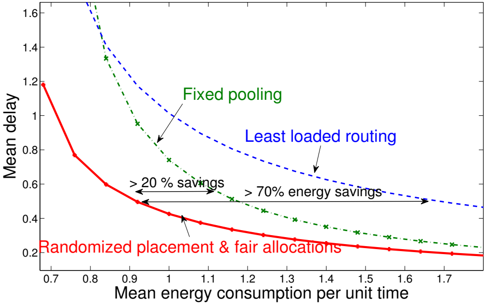
- Parallel service sharing gives substantial performance improvements across loads, e.g., even over least loaded routing
  (Multipath TCP; P2P content delivery $\ddot{\smile}$)

- Back of the envelope performance estimates
  - e.g., $E[D] \propto \frac{1}{c}$, and $E[D] \propto \log\left(\frac{1}{1-\rho/\xi}\right)$

- Enabled evaluation of Performance-Reliability-Energy tradeoffs in engineering CDNs
  - e.g., can limit overlapping of pools for reliability at cost of performance

Energy-Delay Tradeoffs

- Power consumption model: $f(\xi) = \xi^2$ when service rate ξ [Wierman et al. '12].
- Speed scaling policy: Turn server off when idle, turn on with service rate ξ when busy.
- Increasing ξ trades off energy for performance.



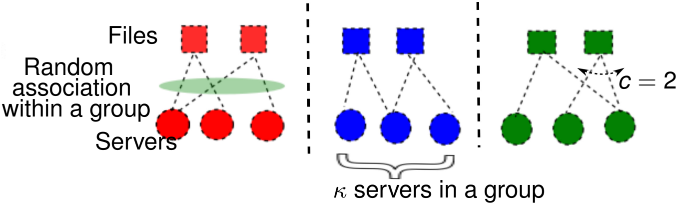
■ Energy-delay tradeoff with varying server speed ξ. $\rho = 0.8$, $\nu = 1$, and $c = 3$.

Reliability Against Correlated Failures

- Consider large scale correlated failures: e.g., about 1% of servers can fail after power outage
- All the *c* copies of some files may be lost
- Recovering from cold storage may incur high fixed costs, less affected by number of files lost [Cidon et al.'13]
- *Goal:* keep probability of a file loss (P_{loss}) low

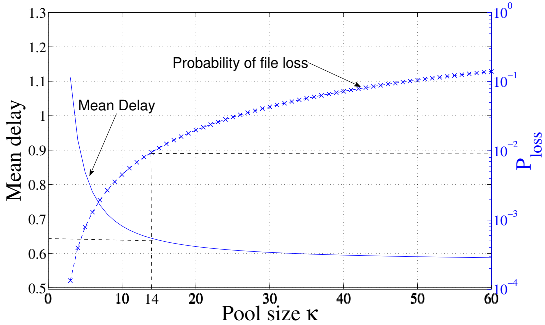
Strategy For Better Reliability

- *File placement policy* [Cidon et al.'13]:
  - Partition set of servers into m/κ pools of size κ
  - Partition set of files into m/κ groups
  - Random file-server association within a group
- Keep κ small for lower P_{loss}



Performance Reliability Tradeoff

- Upon power outage, say with probability 0.01 a server fails.

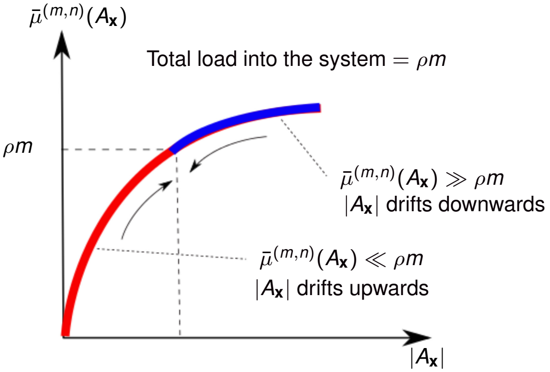


■ $n = 2 \times 10^6$, $m = 400$, $c = 3$, $\rho = 0.7$, and $\nu = 1$.

- At $\kappa = 14$, mean delay is 12% greater than the minimum value, while P_{loss} is less than 1%.
- Decreasing κ can further lower P_{loss} but at the cost of a significant increase in mean delay.

Key Idea behind Asymptotic Result

- In our asymptotic regime, one gets concentration of measure on states $\mathbf{x}$ such that $\bar{\mu}^{(m,n)}(A_{\mathbf{x}}) \approx \rho m$



- Proof a bit technical, uses the exact mean delay expression

More Detailed Intuition for Asymptotic Expression

- In the limiting regime, the invariant distribution concentrates on states $\mathbf{x}$ such that $\bar{\mu}^{(m,n)}(A_{\mathbf{x}}) \approx \rho m$
  - If $\bar{\mu}^{(m,n)}(A) \ll \rho m$ or $\bar{\mu}^{(m,n)}(A) \gg \rho m$, system quickly drifts towards equilibrated states
  - Actual proof quite technical, uses the exact mean delay expression

- Recall:
$\bar{\mu}^{(m,n)}(A) = \xi m(1 - (1 - c/m)^{|A|}) \approx \xi m \left(1 - e^{-c|A|/m}\right)$

- $\bar{\mu}^{(m,n)}(A_x) \approx \rho m$ when $|A_{\mathbf{x}}| \approx \frac{m}{c} \log \left(\frac{1}{1-\rho/\xi}\right)$

- As $n \to \infty$, $\sum_i x_i \approx |A_x|$ w.h.p.

- By Little's Law, $E[D^{(m,n)}] \approx \frac{1}{\lambda c} \log \left(\frac{1}{1-\rho/\xi}\right)$

Balanced Fairness: Defintion

$$r_i(\mathbf{x}) = \frac{\Phi(\mathbf{x} - \mathbf{e}_i)}{\Phi(\mathbf{x})} \quad i = 1, 2$$

- Balanced fair rate allocation is the choice of $\Phi(.)$ such that $\forall \mathbf{x}$, $\mathbf{r}(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}))$ is on the boundary of $\mathcal{C}$.

- Formally, set $\Phi(\mathbf{0}) = 1$, $\Phi(\mathbf{x}) = 0$, $\forall \mathbf{x}$ s.t. $x_i < 0$ for some i, otherwise, set:

$$\Phi(\mathbf{x}) = \inf \left(\alpha : \left(\frac{\Phi(\mathbf{x} - e_1)}{\alpha}, \frac{\Phi(\mathbf{x} - e_2)}{\alpha} \right) \in \mathcal{C} \right)$$