Optimal Closest Policy with QoS and Bandwidth Constraints for Placing Replicas in Tree Networks

Veronika Rehn-Sonigo

GRAAL team, LIP
École Normale Supérieure de Lyon
France

August 2007

Introduction and motivation

- Replica placement in tree networks
- Set of clients (tree leaves): requests with QoS constraints, known in advance
- Internal nodes may be provided with a replica;
  in this case they become servers
  and process requests (up to their capacity limit)

How many replicas required?
Which locations?
Total replica cost?

Introduction and motivation

- Replica placement in tree networks
- Set of clients (tree leaves): requests with QoS constraints, known in advance
- Internal nodes may be provided with a replica;
  in this case they become servers
  and process requests (up to their capacity limit)

How many replicas required?
Which locations?
Total replica cost?

Introduction and motivation

- Replica placement in tree networks
- Set of clients (tree leaves): requests with QoS constraints, known in advance
- Internal nodes may be provided with a replica;
  in this case they become servers
  and process requests (up to their capacity limit)

How many replicas required?
Which locations?
Total replica cost?

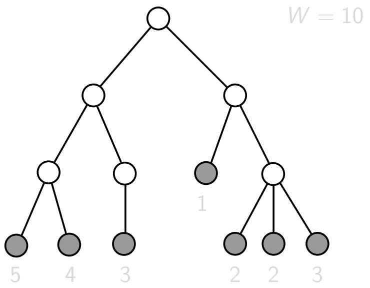
Introduction and motivation

- Replica placement in tree networks
- Set of clients (tree leaves): requests with QoS constraints, known in advance
- Internal nodes may be provided with a replica;
  in this case they become servers
  and process requests (up to their capacity limit)

How many replicas required?
Which locations?
Total replica cost?

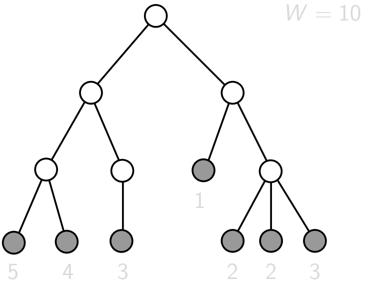
Rule of the game

- Handle all client requests, and minimize cost of replicas
- → Replica Placement problem
- Several policies to assign replicas



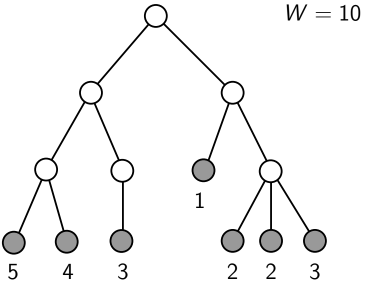
Rule of the game

- Handle all client requests, and minimize cost of replicas
- → Replica Placement problem
- Several policies to assign replicas



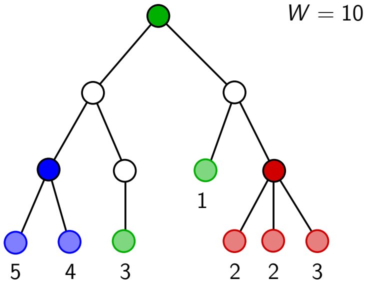
Rule of the game

- Handle all client requests, and minimize cost of replicas
- → REPLICA PLACEMENT problem
- Several policies to assign replicas



Rule of the game

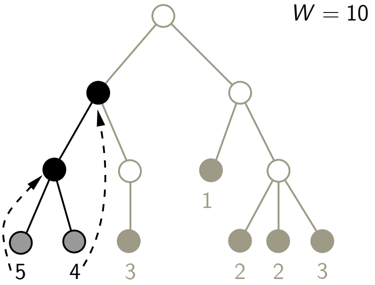
- Handle all client requests, and minimize cost of replicas
- → Replica Placement problem
- Several policies to assign replicas



Closest

Rule of the game

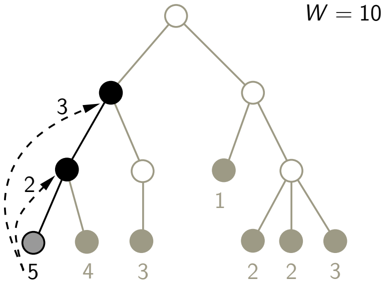
- Handle all client requests, and minimize cost of replicas
- → Replica Placement problem
- Several policies to assign replicas



Upwards

Rule of the game

- Handle all client requests, and minimize cost of replicas
- → REPLICA PLACEMENT problem
- Several policies to assign replicas



Multiple

Outline

1. Framework
2. Complexity results
3. Optimal Replica Placement Algorithm
4. Conclusion

Outline

Definitions and notations

- Distribution tree $\mathcal{T}$, clients $\mathcal{C}$ (leaf nodes), internal nodes $\mathcal{N}$
- **Client** $v \in \mathcal{C}$:
  - Sends $r(v)$ requests per time unit (number of accesses to a single object database)
  - Quality of service q(v) (response time)
- **Node** $j \in \mathcal{N}$:
  - Can contain the object database replica (server) or not
  - Processing capacity W
  - Storage cost sc_j
- **Tree edge:** $l \in \mathcal{L}$ (communication link between nodes)
  - Communication time $comm_l$
  - Bandwidth limit BW_l

Definitions and notations

- Distribution tree $\mathcal{T}$, clients $\mathcal{C}$ (leaf nodes), internal nodes $\mathcal{N}$
- **Client** $v \in \mathcal{C}$:
  - Sends $r(v)$ requests per time unit (number of accesses to a single object database)
  - Quality of service $\mathrm{q}(v)$ (response time)
- **Node** $j \in \mathcal{N}$:
  - Can contain the object database replica (server) or not
  - Processing capacity W
  - Storage cost sc_j
- **Tree edge:** $l \in \mathcal{L}$ (communication link between nodes)
  - Communication time comm_l
  - Bandwidth limit BW_l

Definitions and notations

- Distribution tree $\mathcal{T}$, clients $\mathcal{C}$ (leaf nodes), internal nodes $\mathcal{N}$
- **Client** $v \in \mathcal{C}$:
  - Sends $r(v)$ requests per time unit (number of accesses to a single object database)
  - Quality of service $\mathrm{q}(v)$ (response time)
- **Node** $j \in \mathcal{N}$:
  - Can contain the object database replica (server) or not
  - Processing capacity W
  - Storage cost sc_j
- **Tree edge:** $l \in \mathcal{L}$ (communication link between nodes)
  - Communication time comm_l
  - Bandwidth limit BW_l

Definitions and notations

- Distribution tree $\mathcal{T}$, clients $\mathcal{C}$ (leaf nodes), internal nodes $\mathcal{N}$
- **Client** $v \in \mathcal{C}$:
  - Sends $r(v)$ requests per time unit (number of accesses to a single object database)
  - Quality of service $\mathrm{q}(v)$ (response time)
- **Node** $j \in \mathcal{N}$:
  - Can contain the object database replica (server) or not
  - Processing capacity W
  - Storage cost sc_j
- **Tree edge:** $l \in \mathcal{L}$ (communication link between nodes)
  - Communication time comm_l
  - Bandwidth limit BW_l

Problem instances (1/2)

- Goal: place replicas to process client requests
- Client $i \in \mathcal{C}$: Servers(i) $\subseteq \mathcal{N}$ set of servers responsible for processing its requests
- $r_{i,s}$: number of requests from client i processed by server s $(\sum_{s \in \text{Servers}(i)} r_{i,s} = r_i)$
- $R = \{s \in \mathcal{N} | \; \exists i \in C \; , \; s \in \text{Servers}(i)\}$: set of replicas

Problem instances (2/2)

- Minimize $\sum_{s \in R} sc_s$ under the constraints:
- **Server capacity:** $\forall s \in R, \sum_{i \in \mathcal{C} | s \in \text{Servers}(i)} r_{i,s} \leq W_s$
- **QoS:** $\forall i \in \mathcal{C}, \forall s \in \text{Servers}(i), \sum_{l \in \text{path}[i \to s]} \text{comm}_l \leq q_i$.
- **Link capacity:** $\forall l \in \mathcal{L} \sum_{i \in \mathcal{C}, s \in \text{Servers}(i) | l \in \text{path}[i \to s]} r_{i,s} \leq BW_l$
- Restrict to case where $sc_s = W$:
  Replica Counting problem on homogeneous platforms.

Problem instances (2/2)

- Minimize $\sum_{s \in R} \mathrm{sc}_s$ under the constraints:
- **Server capacity:** $\forall s \in R, \sum_{i \in \mathcal{C} | s \in \mathrm{Servers}(i)} r_{i,s} \leq \mathrm{W}_s$
- **QoS:** $\forall i \in \mathcal{C}, \forall s \in \mathrm{Servers}(i), \sum_{l \in \mathrm{path}[i \to s]} \mathrm{comm}_l \leq \mathrm{q}_i$.
- **Link capacity:** $\forall l \in \mathcal{L} \sum_{i \in \mathcal{C}, s \in \mathrm{Servers}(i) | l \in \mathrm{path}[i \to s]} r_{i,s} \leq \mathrm{BW}_l$
- Restrict to case where $\mathrm{sc}_s = \mathrm{W}$:
  Replica Counting problem on homogeneous platforms.

Problem instances (2/2)

- Minimize $\sum_{s \in R} \text{sc}_s$ under the constraints:
- **Server capacity:** $\forall s \in R, \sum_{i \in \mathcal{C} | s \in \text{Servers}(i)} r_{i,s} \leq \text{W}_s$
- **QoS:** $\forall i \in \mathcal{C}, \forall s \in \text{Servers}(i), \sum_{l \in \text{path}[i \to s]} \text{comm}_l \leq \text{q}_i$.
- **Link capacity:** $\forall l \in \mathcal{L} \sum_{i \in \mathcal{C}, s \in \text{Servers}(i) | l \in \text{path}[i \to s]} r_{i,s} \leq \text{BW}_l$
- Restrict to case where $\text{sc}_s = \text{W}$:
  Replica Counting problem on homogeneous platforms.

Outline

Complexity results

Homogeneous platform: REPLICA COUNTING problem, no bandwidth constraints

	No QoS	**With QoS**
Closest	polynomial [Cidon02,Liu06]	polynomial [Liu06]
Upwards	NP-complete [Be06]	NP-complete [Be06]
Multiple	polynomial [Be06]	NP-complete [Be07]

Heterogeneous platforms: all problems are NP-complete

New result:
Homogeneous platforms with bandwidth and QoS constraints: *Closest* remains polynomial [Re07]

Complexity results

Homogeneous platform: REPLICA COUNTING problem, no bandwidth constraints

	No QoS	**With QoS**
Closest	polynomial [Cidon02,Liu06]	polynomial [Liu06]
Upwards	NP-complete [Be06]	NP-complete [Be06]
Multiple	polynomial [Be06]	NP-complete [Be07]

Heterogeneous platforms: all problems are NP-complete

New result:
Homogeneous platforms with bandwidth and QoS constraints: *Closest* remains polynomial [Re07]

Complexity results

Homogeneous platform: REPLICA COUNTING problem, no bandwidth constraints

	No QoS	With QoS
Closest	polynomial [Cidon02,Liu06]	polynomial [Liu06]
Upwards	NP-complete [Be06]	NP-complete [Be06]
Multiple	polynomial [Be06]	NP-complete [Be07]

Heterogeneous platforms: all problems are NP-complete

New result:
Homogeneous platforms with bandwidth and QoS constraints: *Closest* remains polynomial [Re07]

Outline

1. Framework
2. Complexity results
3. Optimal Replica Placement Algorithm
4. Conclusion

Dealing with Bandwidth Constraints

Optimal *Closest* policy

- Homogeneous platform
- QoS constraints
- Bandwidth constraints

Base: optimal algorithm of Liu et al. for homogeneous data grids with QoS constraints

Provided extensions:

- Bandwidth constraints
- $\mathcal{C} \cap \mathcal{N} = \emptyset$

Dealing with Bandwidth Constraints

Optimal *Closest* policy

- Homogeneous platform
- QoS constraints
- Bandwidth constraints

Base: optimal algorithm of Liu et al. for homogeneous data grids with QoS constraints

Provided extensions:

- Bandwidth constraints
- $\mathcal{C} \cap \mathcal{N} = \emptyset$

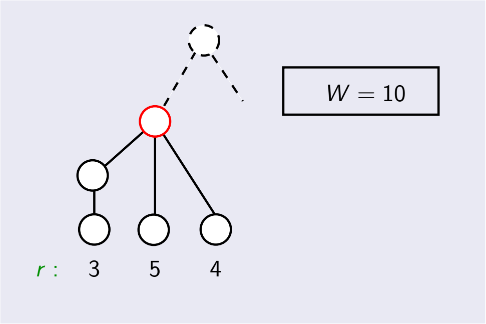
Basic idea:

computation of the minimal necessary number of replicas in a subtree

Basic idea:

computation of the minimal necessary number of replicas in a subtree

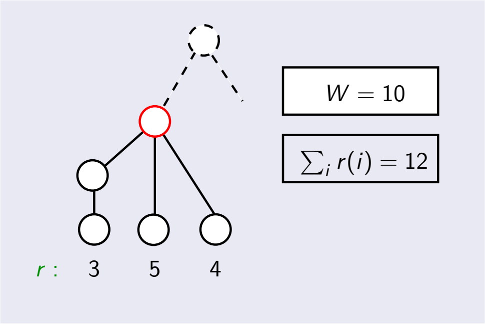
Case 1: too many requests



Basic idea:

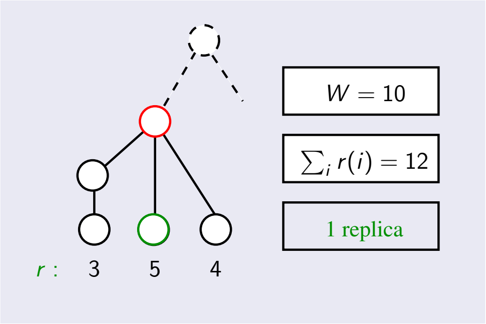
computation of the minimal necessary number of replicas in a subtree

Case 1: too many requests



Basic idea:
computation of the minimal necessary number of replicas in a subtree

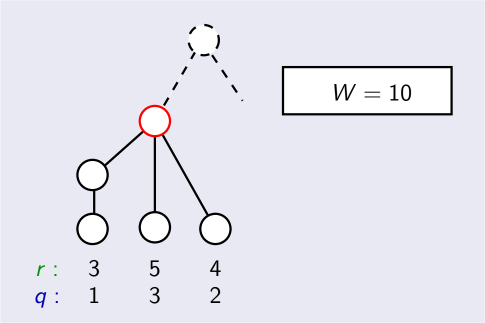
Case 1: too many requests



Basic idea:

computation of the minimal necessary number of replicas in a subtree

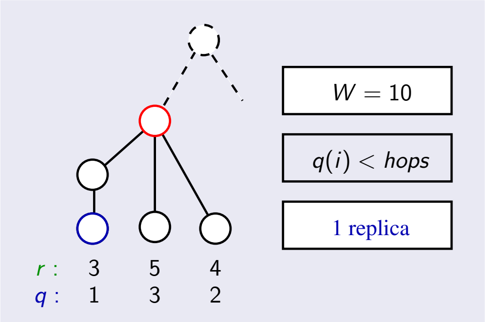
Case 2: QoS constraints



Basic idea:

computation of the minimal necessary number of replicas in a subtree

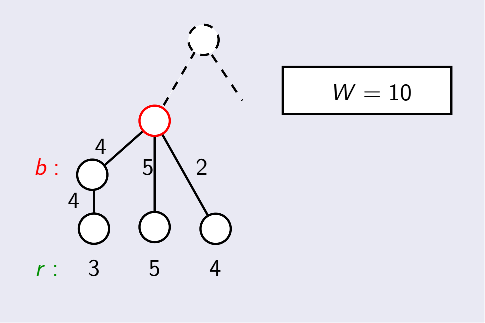
Case 2: QoS constraints



Basic idea:

computation of the minimal necessary number of replicas in a subtree

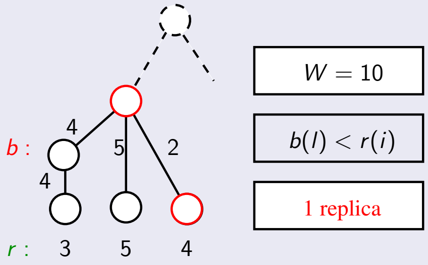
Case 3: bandwidth constraints



Basic idea:

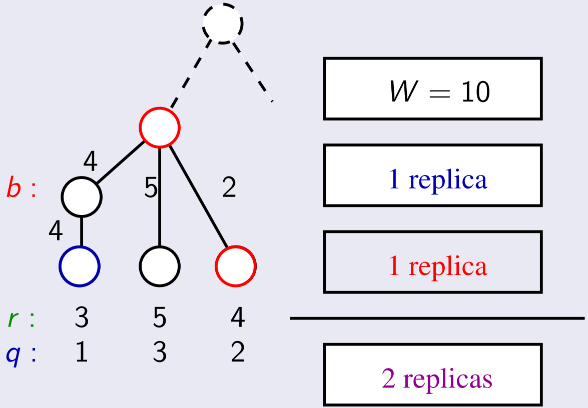
computation of the minimal necessary number of replicas in a subtree

Case 3: bandwidth constraints



Basic idea:

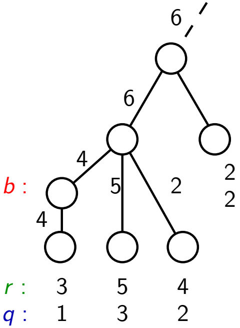
computation of the minimal necessary number of replicas in a subtree



ORP - Optimal Replica Placement Algorithm

Preparation Tree transformation

Step 1 Bottom up computation of the contribution of client requests



$C(v, i)$: the contribution of node v on its i-th ancestor

$e(v, i)$: children of v that have to be equipped with a replica to minimize the contribution on the i-th ancestor of v (respecting some additional constraints).

ORP - Optimal Replica Placement Algorithm

Preparation Tree transformation

Step 1 Bottom up computation of the contribution of client requests

Step 2 Top down replica placement

procedure Place-replica (v, i)

```
if v ∈ C then
    return;
end
place a replica at each node of e(v, i);
forall c ∈ children(v) do
    if c ∈ e(v, i) then
        Place-replica(c,0);
    else
        Place-replica(c,i+1);
    end
end
```

Complexity and Optimality

Theorem

(i) Algorithm ORP runs in polynomial time.

(ii) Algorithm ORP returns an optimal solution to the Replica Placement problem with fixed W, QoS and bandwidth constraints, if there exists a solution.

Outline

1. Framework
2. Complexity results
3. Optimal Replica Placement Algorithm
4. Conclusion

Conclusion

- REPLICA PLACEMENT optimization problem with QoS and bandwidth constraints.
- Restriction to *Closest*/Homogeneous instances
- Polynomial runtime
- Optimality
- Interplay of different-nature constraints

Completion of the study on complexity of *Closest*/Homogeneous.

Future work

- Consider the problem with several object types
- Extension with more complex objective functions

Conclusion

- Replica Placement optimization problem with QoS and bandwidth constraints.
- Restriction to *Closest*/Homogeneous instances
- Polynomial runtime
- Optimality
- Interplay of different-nature constraints

Completion of the study on complexity of *Closest*/Homogeneous.

Future work

- Consider the problem with several object types
- Extension with more complex objective functions