Data Center Services and Optimization

Sobir Bazarbayev
Chris Cai
CS538 October 18 2011
Outline

● Background

● Volley: Automated Data Placement for Geo-Distributed Cloud Services, by Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, Harbinder Bhogan, NSDI '10

● Wide Area Placement of Data Replicas for Fast and Highly Available Data Access, by Fan Ping, Xiaohu Li, Christopher McConnell, Rohini Vabbalareddy, Jeong-Hyon Hwang, DIDC '11

● Dynamic Co-Scheduling of Distributed Computation and Replication, by Huadong Liu, Micah Beck, and Jian Huan, CCGRID '06

● Questions
Data Replica Placement

- Data replication goals:
  - availability
  - reliability
  - load balancing
  - user latency

- Great need for automatically placing data replicas to achieve above goals
Challenges

- Scalability
- Solutions that address operator costs
- Taking advantage of geographically distributed infrastructures
Outline

● Background

● **Volley: Automated Data Placement for Geo-Distributed Cloud Services**, by Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, Harbinder Bhogan, NSDI '10

● **Wide Area Placement of Data Replicas for Fast and Highly Available Data Access**, by Fan Ping, Xiaohu Li, Christopher McConnell, Rohini Vabbalareddy, Jeong-Hyon Hwang, DIDC '11

● **Dynamic Co-Scheduling of Distributed Computation and Replication**, by Huadong Liu, Micah Beck, and Jian Huan, CCGRID '06

● Questions
Volley: Automated Data Placement for Geo-Distributed Cloud Services

- Cloud services span globally distributed datacenters

- Volley
  - automated mechanism for data placement
  - main goals
    - minimize user latencies
    - lower operator costs
  - designed to run on SCOPE (a scalable MapReduce-style platform)
  - evaluated on Live Mesh (cloud storage service)

[Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, Harbinder Bhogan. NSDI, April 2010. Link]
Volley

- Volley is motivated by:
  - shared data, (ex. Google Docs)
  - data inter-dependencies (ex. Facebook)
  - reaching datacenter capacity limits
  - user mobility

- Main Challenges:
  - scalability
  - service operator costs
  - dynamic data sharing
users scattered geographically (Live Messenger)

PLACING ALL DATA ITEMS IN ONE PLACE IS REALLY BAD FOR LATENCY

ALGORITHM NEEDS TO HANDLE USER LOCATIONS THAT CAN VARY

Volley Design

- Applications provide following information to Volley:
  - processed requests log
  - requirements on RAM, disk, and CPU per transaction
  - a capacity cost model
  - a model of latency between datacenters and between datacenters and clients
  - current locations of every data item
Volley Algorithm

Phase 1: Compute Initial Placement
● calculate geographic centroid for each data based on client locations, ignore data-interdependencies

Phase 2: Iteratively Move Data to Reduce Latency
● refine centroid for each data iteratively
  ○ consider client locations and data-interdependencies
  ○ use weighted spring model and spherical coordinate system
Phase 3: Iteratively Collapse Data to Datacenters

- modify data placement to satisfy data center capacity constraints
- for datacenters over their capacity, move least accessed items to next closest data center
Volley Evaluation

- Live Mesh from June 2009
  - 12 geographically diverse datacenters
  - month long evaluation
  - analytical evaluation using latency model (Agarwal SIGCOMM’09)
    - based on 49.9 million measurements across 3.5 million end-hosts
  - live experiment using Planetlab clients
Volley Evaluation

- Results compared to following heuristics
  - **commonIP**
    - place data closest to IP that most commonly accessed it
  - **oneDC**
    - put all data in one datacenter
  - **hash**
    - hash data to datacenters so as to optimize for load-balancing
user transaction latency (analytic evaluation)

INCLUDES SERVER-SERVER (SAME DC OR CROSS-DC) AND SERVER-USER

user transaction latency (ms)

<table>
<thead>
<tr>
<th>Percentile</th>
<th>hash</th>
<th>oneDC</th>
<th>commonIP</th>
<th>volley</th>
</tr>
</thead>
<tbody>
<tr>
<td>50th</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75th</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95th</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

percentile of total user transactions
inter-DC traffic (analytic evaluation)

WAN traffic is a major source of cost for operators.

- Volley: real money
- commonIP: 0.2059
- Hash: 0.7929
- oneDC: 0.0000

fraction of messages that are inter-DC

Volley Summary

- Volley solves the following problems
  - partition user data across multiple data centers
  - reduce operator cost and user latency

- Main results from LiveMesh evaluation
  - reduces datacenter capacity skew by over 2x
  - reduces inter-DC traffic by over 1.8x
  - reduces user latency by 30% at 75th percentile

Outline

- Background

- **Volley: Automated Data Placement for Geo-Distributed Cloud Services**, by Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, Harbinder Bhogan, **NSDI '10**

- **Wide Area Placement of Data Replicas for Fast and Highly Available Data Access**, by Fan Ping, Xiaohu Li, Christopher McConnell, Rohini Vabbalareddy, Jeong-Hyon Hwang, **DIDC '11**

- **Dynamic Co-Scheduling of Distributed Computation and Replication**, by Huadong Liu, Micah Beck, and Jian Huan, **CCGRID '06**

- Questions
Wide Area Placement of Data Replicas for Fast and Highly Available Data Access - Ping, et al.

● Similar goal as Volley
● Volley:
  ○ automated data placement considering data interdependencies
  ○ minimize user latency and operation cost
● Wide Area Placement:
  ○ focuses on data replica placement
  ○ maximize user defined objective function
Problem Statement

● Given a target degree of replication $r$, choose $r$ out of all possible locations to store the replicas such that a system optimization goal is achieved.

● System optimization is based on a user-defined objective function, e.g. $o(d,a) = 1/d$ or $o(d,a) = 1/(d \times \text{sqrt}(1-a))$
Replication Location Selection Algorithm

- Basically brute force
- Considering all possible $r$ locations out of all candidates
- But can do this efficiently (why?)
Cluster the clients in the coordinate system

[Subsequent graphs are borrowed from Fan Ping, et al link]
Centroids of the clusters
Evaluation

- Results compared to following replica placement strategies
  - *Random*
    - choose $r$ locations randomly
  - *Optimal [delay] or [availability] or [delay, availability]*
    - calculate replica locations assuming we know the future data access (infeasible)
  - *Most Available Nodes*
    - choose $r$ locations with highest availability
  - Optimized [delay] or [availability] or [delay, availability]
    - using the replica location selection algorithm we discussed
Average Access Delay vs Number of Replicas

The graph shows the relationship between the average access delay (in milliseconds) and the number of replicas, ranging from 2 to 7. The y-axis represents the average access delay, while the x-axis shows the number of replicas. Various strategies and optimizations are compared, including:

- **random**
- **optimal [delay]**
- **optimal [availability]**
- **most available nodes**
- **optimized [delay]**
- **optimized [availability]**
- **optimized [delay, availability]**

Each strategy has a distinct line on the graph, allowing for a clear comparison of performance across different numbers of replicas.
Availability vs Number of Replicas
Outline

● Background

● Volley: Automated Data Placement for Geo-Distributed Cloud Services, by Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, Harbinder Bhogan, NSDI '10

● Wide Area Placement of Data Replicas for Fast and Highly Available Data Access, by Fan Ping, Xiaohu Li, Christopher McConnell, Rohini Vabbalareddy, Jeong-Hyon Hwang, DIDC '11

● Dynamic Co-Scheduling of Distributed Computation and Replication, by Huadong Liu, Micah Beck, and Jian Huan, CCGRID '06

● Questions
Dynamic Co-Scheduling of Distributed Computation and Replication

- Volley
  - high data interdependency
  - automate data placement to minimize user latency and operator cost

- Wide Area Placement
  - *k-way* replicated data for lower latency and higher availability
  - automate replication placement to maximize user defined objective function (latency, availability)

- This paper addresses data movement in different environment
  - considers data movement in both computation and data intensive distributed system
  - similar goal as last two papers of minimizing latency
Dynamic Co-Scheduling of Distributed Computation and Replication

- Dynamic co-scheduling algorithm that integrates scheduling of computation and runtime data movement

- Goals:
  - improve server utilization
  - minimize latency

- Overview
  - adaptively measures server performance
    - computation power
    - data transfer rate
  - dynamically assign tasks to servers
  - direct data movement between servers

[Huadong Liu, Micah Beck, and Jian Huan. Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID), 2006.]
Adaptive Scheduling of Computation

- Assign as many tasks to fast servers, avoid being stalled by slow or faulty servers

- Three generic mechanisms
  - Dynamically ranked pool of servers
  - Two level priority queue of tasks
  - Competition avoidant task management scheme

![Figure 2: A snapshot of the two-level priority queue](Dynamic Co-Scheduling of Distributed Computation and Replication, Page 4)
Dynamic Scheduling of Replication

- Fast servers can help slow servers by repeating tasks on replicas
- Some partitions only reside on set of slow servers
- Can move partitions to fast servers if time spent on data movement does not exceed the profit that we gain from migrating the task
Figure 8: Execution time of volume rendering with different $w$
Summary

● Maximizing server utilization and minimizing application execution time by co-scheduling algorithm

● Greatly improving both server utilization and application performance even with a small number of replicas
Outline

● Background

● Volley: Automated Data Placement for Geo-Distributed Cloud Services, by Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, Harbinder Bhogan, NSDI '10

● Wide Area Placement of Data Replicas for Fast and Highly Available Data Access, by Fan Ping, Xiaohu Li, Christopher McConnell, Rohini Vabbalareddy, Jeong-Hyon Hwang, DIDC '11

● Dynamic Co-Scheduling of Distributed Computation and Replication, by Huadong Liu, Micah Beck, and Jian Huan, CCGRID '06

● Questions
Questions

● As distributed datacenters are becoming more and more geographically distributed how important is it to develop good techniques to automate data placement?

● When are the applications good candidates for data replica placement algorithms?

● Which techniques we have described fit following applications?
  ○ Google Docs
  ○ Facebook status updates and wall posts
  ○ Flickr