Provisioning Data-Intensive Workloads on a Cloud

P. Martin, R. Mian
School of Computing

J. L. Vázquez-Poletti
DSA Research Group
Facultad de Informática

Outline

- Data-intensive applications and the cloud
- The data provisioning problem
- Data partitioning
- Partition assignment
- Future work

Data-Intensive Workloads

- OLTP
  - Read/update; large number of requests; small data/request; inter-request parallelism
- Business analytics
  - Read; small number of requests; large data/request; intra-request parallelism
- MapReduce
  - Read/update; small number of requests; large data/request; intra-request parallelism
- Social computing
  - Read; large number of requests; small data/request; high level of interconnection in data; inter-request parallelism

Cloud offers scalability, elasticity, parallelism!

Challenges to Provisioning in the Cloud

- Dynamically exploiting the scalability and elasticity of the Cloud without a large jump in complexity
- Placing the data in the Cloud to exploit the tradeoff of *locality* and *replication*
- Effectively managing the data in the Cloud
  - Maintaining consistency of replicas
  - Adapting to shifts in workload patterns

Data Provisioning Problem

Given a set of data objects $D = \{d_1, d_2, \ldots, d_n\}$, and an application W with requests $R = \{r_1, r_2, \ldots, r_m\}$ determine a placement of the data in D on Virtual Machines (VMs) such that the SLO's of the requests in R are satisfied and the cost of using the resources of the (public) Cloud are minimized.

- Two parts to solving the problem
  - Data partitioning
  - Partition assignment

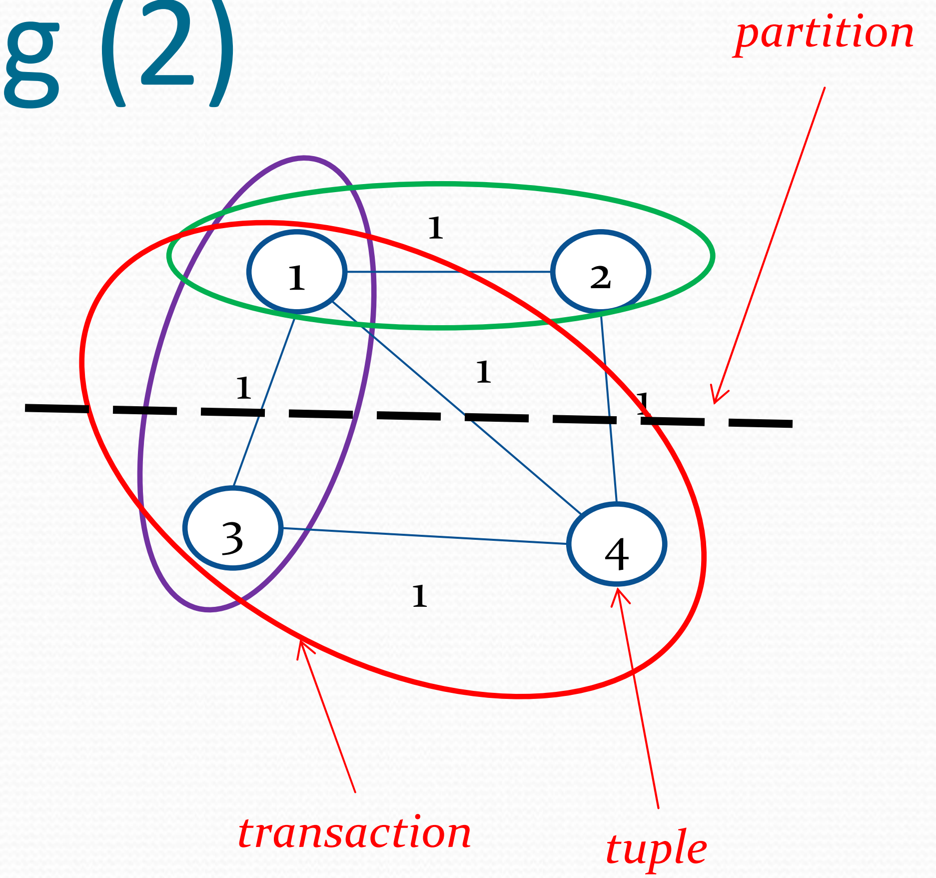
Data Partitioning

Determine the partitioning of each d_i in D that best exploits the *locality* inherent in the workload of application W.

- Top-down approach
  - Traditional distributed database design
  - Hash / range partitioning

Data Partitioning (2)

- Bottom-up approach
  - Data-driven – extract partitions from how data is used
  - Eg. Schism uses a graph partitioning scheme
    - Minimize number of distributed transactions while balancing data on nodes
    - Can also consider replication



C. Curino, E. Jones, Y. Zhang and S. Madden. Schism: A Workload-Driven Approach to Database Replication and Partitioning, *Proc of 36th International Conference on Very Large Data Bases*, September 13 - 17, 2010, Singapore.

Data Partitioning (3)

- Bottom-up approach is *workload-aware*
  - Partitions based on how data actually used
  - Can adapt partitions to changes in workload
- Tuples are basic unit considered in Schism so need approaches for scalability
- Does bottom-up approach suit various types of data-intensive workloads?
  - Good for inter-request parallelism (OLTP, social computing
  - Combine with hash/range partitioning for intra-request parallelism of MapReduce and OLAP?

Partition Assignment

Partitions are assigned to VMs such that a VM can satisfy SLOs of requests it must process and the total cost of the configuration is minimized.

- Reactive versus predictive assignment methods
  - Reactive method is simpler but would take longer to converge – can be heuristics based
  - Predictive is more complex but potentially more effective – must account for concurrency and contention for resources on a VM

Cost Model

$$Cost(C) = R(C) + \sum_{t \in T} P_t(C) \quad (\$ / hour)$$

- *R(C)* – resource costs of configuration *C*
  - VM, data access and storage costs
  - Contention on a VM modeled with a QNM
- $P_t(C)$ – penalty costs for request class *t* on *C*
  - Cost associated with under-provisioning resources
  - Cost ($) / hour that requests of class *t* are under-performing

The Cloud Billing Rates

Prevailing Billing Rates

Google

Resource	Unit	Unit cost
Outgoing Bandwidth	gigabytes	$0.12
Incoming Bandwidth	gigabytes	$0.10
CPU Time	CPU hours	$0.10
Stored Data	gigabytes per month	$0.15
Recipients Emailed	recipients	$0.0001

Source: Google Code

amazon.com

Resource	Unit	Unit Cost
Data Transfer-in	gigabytes	$ 0.10
Data Transfer-out	gigabytes	$ 0.14
Storage	gigabytes/month	$ 0.15
CPU Compute Time	Instance hours	$ 0.125

Source: Amazon, Amazon

Microsoft

Resource	Unit	Unit Cost
Data Transmissions-in	gigabytes	$ 0.10
Data Transmissions–out	gigabytes	$ 0.15
Storage	gigabytes/month	$ 0.15
Compute Time	Machine Hours	$ 0.12
Storage Transactions	10K Application Requests	$0.01

Source: Microsoft Azure

netmagic

When it's mission critical

Resource	Unit	Unit Cost
CloudNet (Basic cloud service operation)	Rs/month	7000
CloudServe (On-Demand Server Provisioning)	Rs/month	10,000
Private Cloud	Rs/month	20,000

Source: BusinessWorld

Machine Type	Cores	C.U.	Memory	Storage	Platform
Standard On-Demand Instances					
Small (Default)	1	1	1.7GB	160GB	32bit
Large	2	2	7.5GB	850GB	64bit
Extra Large	4	2	15GB	1,690GB	64bit
High CPU On-Demand Instances					
Medium	2	2.5	1.7GB	350GB	32bit
Extra Large	8	2.5	7GB	1,690GB	64bit

amazon web services™

Different availability zones

Machine Type	Price in USA
Standard On-Demand Instances	
Small (Default)	$0.10/hour
Large	$0.40/hour
Extra Large	$0.80/hour
High CPU On-Demand Instances	
Medium	$0.20/hour
Extra Large	$0.80/hour

Amazon Simple Storage Service

Pricing

United States

Storage

$0.150 per GB - first 50 TB / month of storage used
$0.140 per GB - next 50 TB / month of storage used
$0.130 per GB - next 400 TB /month of storage used
$0.120 per GB - storage used / month over 500 TB

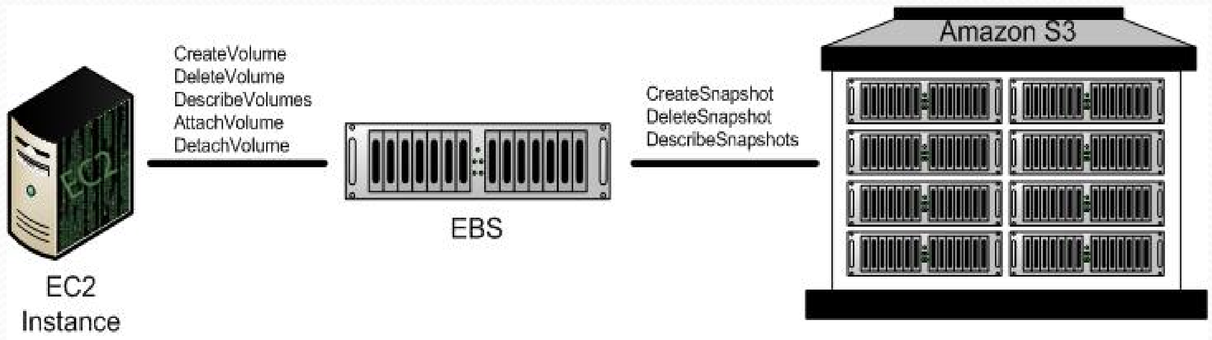
Data Transfer

$0.100 per GB - all data transfer in
$0.170 per GB - first 10 TB / month data transfer out
$0.130 per GB - next 40 TB / month data transfer out
$0.110 per GB - next 100 TB / month data transfer out
$0.100 per GB - data transfer out / month over 150 TB

Requests

$0.01 per 1,000 PUT, COPY, POST, or LIST requests
$0.01 per 10,000 GET and all other requests*

*There is no charge for delete requests



Configuration Selection

- We search space of possible configurations using ***tabu search*** algorithm
- From any given configuration the possible "moves" include
  - Upgrade the VM with heaviest load
  - Add a new VM and shift class from VM with heaviest load
  - Shift a class from a heavily loaded to a lightly loaded VM
  - Merge two lightly loaded VMs

Future Work

- Complete initial evaluation of partition assignment
- Extend the cost model
  - replica costs, communication costs, distributed transactions, license fees, different availability zones
- Data partitioning
  - Develop bottom-up approach that combines Schism graph partitioning and hash/range partitioning to support range of data-intensive applications

Future Work (2)

- Adaptable data placement
  - Include a feedback loop to monitor performance of configuration and adapt resource provisioning and data partitions when workload changes
  - Key challenges are
    - Detecting workload shifts
    - Determining cost-effective changes
    - Minimizing the impact of making changes

Grazie!