Rapid Processing of Synthetic Seismograms using Windows Azure Cloud

Vedapraphak Subramanian, Liqiang Wang
Department of Computer Science
University of Wyoming

En-Jui Lee, Po Chen
Department of Geology and Geophysics
University of Wyoming
• Scientific applications
  – Large amount of computing resources
  – Massive storage for datasets

• Traditionally Handled by HPC Clusters
  – But Cost-ineffective

• Claim: Cloud Computing is a better substitute
• Conduct numerical simulation of synthetic seismograms

• Implemented a system on Azure cloud
  – to generate synthetic seismograms on the fly based on user queries
  – to deliver them in real-time
Synthetic Seismograms

- Seismogram is a record of the ground shaking.
- Synthetic seismograms are generated by solving seismic wave – equation.
- Method is **rapid centroid moment tensor (CMT) inversion method**
  - Based on 3D velocity model
- To increase efficiency, we store Receiver Green Tensors strain fields
  - Generated at the receiver’s location
The input parameters for this wave-equation are
- latitude, longitude and depth of the earthquake locations
- strike, dip, and rake (source parameters)
Why synthetic seismograms is important?

- Helps to map the Earth’s internal structure
- Locate and measure the size of different seismic sources
- For realistic interpretation of structures
- Seismic Hazard Analysis
Windows Azure
Windows Azure

• Windows Azure is Platform as a service architecture
• Provides
  – Scalable cloud operating system
  – Data storage system
• User controls the hosted application
• User cannot control the underlying infrastructure
Service Architecture

- Azure service consists of:
  - **Web role**
    - For web application
    - For user interface
  - **Worker role**
    - For generalized development
    - For background processing
- Roles are virtual machines
- Say, 2 instances of worker role = 2 virtual machines running the code of worker role
Storage Abstractions

• Blobs
• Tables
• Queues
• Drives
• Our system uses only first three
Blob Storage

- **Blobs**
  - Interface for storing files
  - Two types namely Page blobs and Block blobs
  - Containers for grouping
**Table Storage**

- **Tables**
  - Structured storage
  - Consists set of entities, which contain a set of properties
  - Partition key and Row key

![Diagram showing Table Storage](image)
Queue Storage

- Queues
  - Reliable storage and delivery of messages
  - Communication between roles

Account
  - Geo

Queue
  - Orders

Message
  - ID = ....
  - ID = ...
• **Azure master system**
  – Automatically load balance based on the partition key

• **Partition key for various storage abstractions**
  – Blobs – Container Name + Blob Name
  – Entities – Table Name + Partition Key
  – Messages – Queue Name
Implementation of the System
Overview of the System

- The architecture of the system:
  - Web Role
    - User interface
  - Job Manager
    - Coordinate the computation
    - Monitor the system

User Request Queue

Computation Input Queue

Computation Output Queue

Windows Azure Storage (Blob, Table, Queue)
– Computation Worker Role
  • Computation stuff
– 3 Azure Queues
  • Communication interface between the roles
- **Receive** the user input
- **Place** the request as message in request queue

User ➔ Web Role ➔ Request Queue ➔ Windows Azure Storage (Blob, Table, Queue)
• **Retrieve** the message from request queue
• **Read** the job
• **Divide** the job into sub-jobs
• **Place** the sub-jobs as message in input-computation queue
Coordinate the Computation

- Num of computation worker roles: 5
- Num of CPU cores in each instance: 8
- Input: 1000 source locations
- Num of queue messages: 1000 / 8 = 125
- 125 queue messages served by 5 computation worker roles
- Performance gain factor (Theoretical)
  
  \[ \text{Performance gain factor} = \text{Num of CPU cores} \times \text{Num of instances} \]
  
  \[ = 8 \times 5 = 40 \]
Inside Computation Worker Role

- Retrieve the message
- **Process** the sub-jobs in parallel using .NET Task Parallel Library
- Write the result
- Send message stating job completed

**Windows Azure** Storage (Blob, Table, Queue)
Monitor System Response

• Based on the response time of the message

• Threshold response time is 2ms

• If exceeds
  – Allocate a new instance of computational worker role
  – Maintain its detail
De-allocate if
- any new instance has been allocated
- and its lifetime > one hour
- and no message in the queue
Monitor System Response

Legend
- Response time

Distributed File System

Computation Input Queue

Worker
Worker
Worker
Worker

Job Manager

Request to allocate VM
Job Manager

- Allocation & de-allocation are asynchronous

- So, mutual exclusion lock are enforced between
  - Job assignment
  - De-allocation of an instance
Data Storage

- Seismic wave observations stored as data file
- Each data file is represented by its own latitude and longitude
- So, blob name = latitude + longitude
- For grouping the blobs, the region of California is divided into blocks
  - based on the seismic wave observation stations
Currently 4096 stations = 4096 blocks

Each block is characterized by range of latitude and longitude

So, block identification number = range of latitude + longitude

Container name = identification number
• **Divide** California region into 16 parts

• **1 Table for 1 part**

• **Store** list of blocks in the part into the table

• **Helps** in better retrieval
- **Locate** blob corresponding to the given point
  - Retrieve the container name and blob name

  - Retrieve container name from the table
    - Locate the table
    - Do a linear search inside the table

  - Blob name = latitude + longitude
Data Query Algorithm
Experiment
Performance was evaluated on

- various configurations and number of instances of computational worker role

- datasets from different number of seismic wave observation
Performance Measurement – Execution Time

- Single Worker (4 core)
- Four Worker (4 core)
- Single worker (4 core) + TPL
- Four Worker (4 core) + TPL
- Two worker (8 core) + TPL

Execution Time (seconds) vs. Number of stations
Conclusion
Conclusion

• Implemented the system on Windows Azure

• Hence Cloud is a better substitute

• Future Work:
  – Add Seismic Hazard Analysis feature to the system
Thanks