Scientific Tool for Applications Harnessing the Cloud (STAHC)

Jonathan C. Rann
North Carolina A&T State University
Research Alliance in Math and Science
Computer Science and Mathematics Division, Oak Ridge National Laboratory
Email: jcrann84@gmail.com Mentor: Rob Gillen
http://sites.google.com/a/g.ornl.gov/j_rann/

LOCAL MACHINE COMPUTING

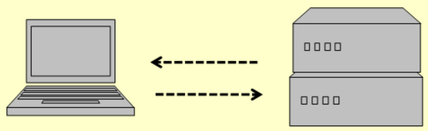


Figure 1. Local machine computing.
Client accesses local resources from client-side hardware. Since the resources are both fixed and shared, users may have to wait until they are free

CLOUD COMPUTING

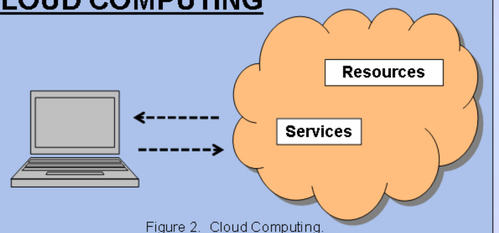


Figure 2. Cloud Computing.
Client accesses resources and services over the internet. Those resources and services can grow and/or shrink as needed.

RESEARCH OBJECTIVE

To provide a mechanism by which to deploy similarly characterized scientific applications to the cloud

BACKGROUND

Researchers here at the laboratory perform a wide array of calculations and computations on local machines which can often be time consuming and expensive. Cloud computing offers a low cost means of horizontal scaling, so it can possibly analyze data more efficiently.

MACHINE SCALING

Horizontal scaling, a big benefit of cloud computing, occurs when the number of machines used increases. Vertical scaling, on the other hand, occurs when more resources are added to a single machine.

THE CLOUD

- online-based method used to access shared resources
- provides elasticity, dynamicism, and on-demand services
- generally provides services at low cost
- ideal when used for temporary or 'burst-style' loads

MICROSOFT WINDOWS AZURE

- application platform based in the cloud
- platform provided as a service (PaaS)
  - little knowledge of underlying infrastructure technologies
- allows applications to be hosted and run at Microsoft datacenters
- allows development, management, and hosting of applications.
- platform API built on REST, HTTP, and XML
- Azure storage consists of blobs, tables, and queues
  - blobs are most simple way to store data in the cloud
  - queues provide means of communications within Azure
- Azure worker role performs background work for cloud application
  - reads, executes, then clears instructions from queue

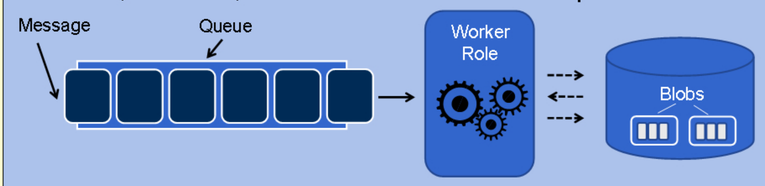


Figure 3. Message Queue, worker role, and blob storage.
Messages are sent to the queue, then to the worker role, which pulls needed data from blob storage. When work is complete, data is sent back to blob storage.

STAHC

- application that deploys scientific applications to the cloud
- composed of a deployment manager and worker applications

DEPLOYMENT MANAGER

- accepts command line arguments which are used in methods
- methods include:
  - upload files
    - instantiates cloud blob client
    - creates instance of blob for configuration and package files
    - uploads files
  - create, start, stop, and delete deployment
    - sends XML requests to cloud platform, waits for response

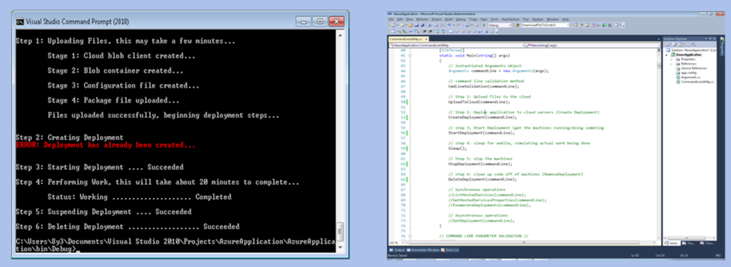
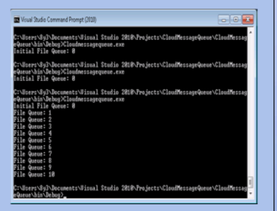


Figure 4. Deployment Manager
The command line deployment manages application helps to Manage applications deployed to the cloud.

WORKER APPLICATION

- queue and message creator code
  - creates message
    - messages consist of instructions
  - puts message in queue
- worker role
  - uses loop to check queue for messages
  - if message not found,
    - waits and checks again
  - if message found
    - takes message from queue
    - performs work
  - once work is completed
    - old message is deleted
    - checks for new message



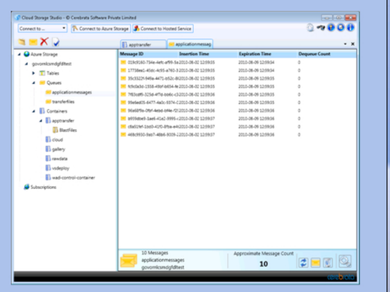


Figure 5. Message Queue
The message queue code creates messages for the worker to process.

SCENARIO: BLASTn

- Basic Local Alignment Search Tool (BLAST)
- maintained by the National Center for Biotechnology Information
- compares nucleotide queries and database sequences
- 'match' or 'hit' is made when database sequence resembles query

RESULTS
- successfully executed blast within the cloud
- validated that cloud deployment application functioned correctly

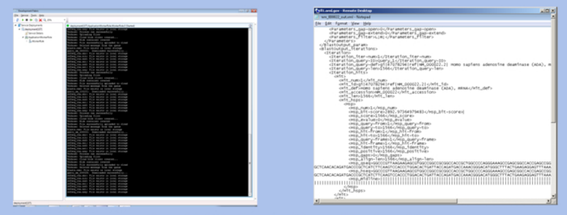


Figure 6. Cloud application and blast results.

SCENARIO: rMiner

- created by ORNL's Dr. Raju Vatsavai
- calculates the Normalized Vegetation Density Index (NVDI)
  - data taken from multi-band satellite images

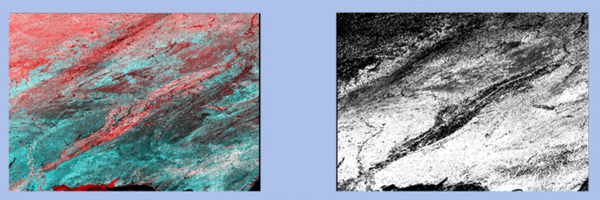


Figure 7. rMiner images
rMiner takes multilayered satellite imagery (left) and performs a calculation at a pixel by pixel level to generate the Normalized Vegetation Density Index (right).

RESULTS
- horizontally scaled to 18 nodes
- processed 1000 images in less than 3 hours
- generated 85.5 GB of imagery
- **demonstrated ability to run non-Microsoft code**
  - rMiner written in Java

SIGNIFICANCE

- applications horizontally-scale easily
- applications require **no change to code** for deployment
- users need little knowledge of background processes
- runs non-Microsoft code

The Research Alliance in Math and Science program is sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy. The work was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725.
This work has been authored by a contractor of the U.S. Government, accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.
I would like to thank Rob Gillen for the opportunity to work on this project. Also, I would like to thank Dr. Anna Yu and Dr. Gregory Dozier for their continued support. Finally, special thanks to Debbie McCoy, who made provisions for this research experience.