PRINCETON CONSULTANTS
Information Technology and Management Consulting

AWS First Annual EC2 Spotathon Submission

November 2012

What is your Spot application, what problem does it solve, and why is it important? For example: If you are representing a company or organization, what does your company do and how does your Spot application fit in? [1 page max]

High-frequency stock trading (HFT) is one of the very few places in finance where tiny start-ups can compete against giant institutions on a level playing field.

No human being can trade effectively at HFT speeds. The objective is to create computer programs that can rapidly assess market conditions, then offer to sell shares to buyers, and buy shares from sellers.

The differentiator between competitors is the quality of the algorithms behind their HFT trading programs, and what drives better algorithms is research.

One challenge of this research is the sheer quantity of data. Another is the constant need to tune and devise new algorithms. Towards this end, it is not unusual for a large financial institution to purchase and maintain expensive dedicated supercomputer farms for its quantitative research.

The typical cycle of a research project is that proposed Research Ideas are discussed, approved, coded, then "backtested" against prior market and related data using different combinations of parameters. For a single research project, backtesting many involve reading hundreds of TB data, and require tens of thousands of hours of processing.

Using AWS spot instances as a backbone, our consulting firm has been able to help a start-up hedge fund cost-effectively perform research and devise algorithms that trade hundreds of millions of dollars per day competing successfully against some of the largest private funds in the world.

How you have incorporated Amazon EC2 Spot Instances into your application? Please describe your application architecture, including: how you evaluate the Spot market, how you bid on and manage your Spot Instances, how you handle Spot interruptions, how you integrate them with On Demand or other computing resources (if any), and any third party architecture or software you use. [1 page max]

We started by completely refactoring the HFTs research away from massive, multi-hour jobs into hundreds of smaller pieces. After evaluating multiple existing multi-processing controller frameworks, we created our own that we dubbed OptiSpotter.

An OptiSpotter user maps which pieces can run concurrently, and which need to be gated. OptiSpotter then marshalls these pieces into thousands of small sub-jobs, each with a specific calculation for a specific region of data.

OptiSpotter then places these jobs into an SQS queue. The choice of queue depends on the memory and I/O requirements of the sub-job. For each queue there exists several possible instance types in different availability-zones that satisfy the sub-job requirements.

Using techniques similar to the ones used for the HFT market trading itself, the system monitors the dynamics of the SQS queues, the recent spot price history in several Availability Zones and instance performance. Using custom optimization algorithms, OptiSpotter determines the most efficient collection of spot instances to promptly process the queue contents for the least cost. In rare cases the optimal cluster may include some On Demand instances if current spot prices are highly volatile. Load can be automatically moved to other Regions if cost merits it.

What cost savings do you achieve by using Spot Instances in your application? For example: How many instance-hours does your application use, how many are on Spot, and what is the total cost of running your Spot application? What would the total cost be if you were not using Spot Instances? What percent savings do you achieve? [1/2 page max]

Due to the nature of the applications, the demand on OptiSpotter is highly variable over time. However, total instance-hours per month regularly exceed 10,000 of which almost all are on spot instances.

By moving the load from On Demand instances to the price optimized spot cluster the cost of running the system has been reduced by 90%.

What performance benefit(s) does your Spot application achieve by using Spot Instances? Please describe. For example: Are you able to achieve shorter time to results because you can deploy more EC2 instances? If you're running a simulation, does Spot enable you to execute more computational runs to improve the accuracy of your solution? [1/2 page max]

Levering the unique advantages of AWS Spot Instances has enabled the hedge fund to greatly speed up its research, improving turnaround time by 5x-10x. Research Ideas formally requiring overnight runs are now available within an hour. This is a critical competitive advantage in today's fast-moving market conditions.

The significant cost savings with Spot means that the group can research ten times as many ideas for the same cost -- providing better and more rigorous algorithms for its investors.

Finally, the low administrative overhead in general of an EC2 cluster, combined with automatic monitoring of our OptiSpotter over the Spot Instances saves valuable staff time for the hedge fund's small IT staff, allowing them to focus their time on the truly proprietary aspects of their trading infrastructure.

What computational scale have you been able to achieve with your Spot application? For example: What is the most number of concurrent instances you have been able to run? Does your application run across many regions and instance types? How many instance-hours does it (did it) take to run your application? [1/2 page max]

The OptiSpotter optimizer builds and maintains the most cost effective spot cluster to complete the current workload in the required time. But the definition of "optimal" can vary by circumstance.

In stock trading, a Fund Manager will tell the trading desk to "purchase X shares with Y urgency". If the number of shares are large, this trade has the potential of having high "market price impact." Wanting to buy a huge number of shares rapidly will drive the prices up. One potential mitigator is to reduce urgency. If the Fund Manager specifies "low urgency" then the trader can use the fluctuations of prices to buy when the prices swings lower. It also lessens the adverse price impact of a large short term demand.

Using a similar strategy, when a research asks to run a series of jobs representing a Research Idea, OptiSpotter wants to know the overall size of the request, and its urgency. It will vary its Spot requests and pricing accordingly. In some cases, for instance at the end of a workday, the researcher may say "get this done

by tomorrow 7AM." In other cases, the researcher may specify ASAP. This not only helps reduce costs for the fund, but provides a external benefit of having low urgency tasks look for lulls, thereby helping overall network balance for all users.

In the fall of 2012, it has been typical run up to a few hundred spot instances concurrently across a number of availability zones and using a mix of instance types. Total processing time for a single Research Idea varies greatly. The average Idea requires 500 instance-hours to run. Due to the success of Optispotter, the number of ideas being run has doubled every month during the fall of 2012.

For more information, Contact

Steve Sashihara
CEO
Princeton Consultants, Inc
Email: SSashihara@princeton.com
Web: www.princeton.com