

Automaton: A Self-optimizing Cloud Server

Supreeth Shastri and David Irwin University of Massachusetts Amherst



PROBLEM STATEMENT

IaaS is evolving into a complex marketplace

	Price	Performance	Availability
On-demand	Fixed	Fixed	Fixed
Burstable VM	Fixed	Variable	Fixed
EC2 Spot	Variable	Fixed	Variable
EC2 Spot-blocks	Variable	Fixed	Fixed
Google Preemptible	Fixed	Fixed	Variable

But applications are **designed** to run on servers with *immutable* characteristics

- Transient Servers** are inherently dynamic:
- Server characteristics change in real-time
 - Failures are frequent but preceded by a warning

Prior works (i) try to predict market behavior (ii) build centralized third-party services to multiplex applications

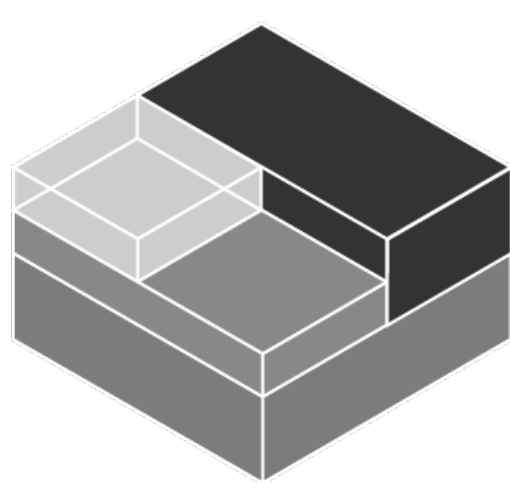
How to enable any unmodified application to exploit cloud market dynamics?

AUTOMATON OVERVIEW

Separate server’s management plane from usage plane

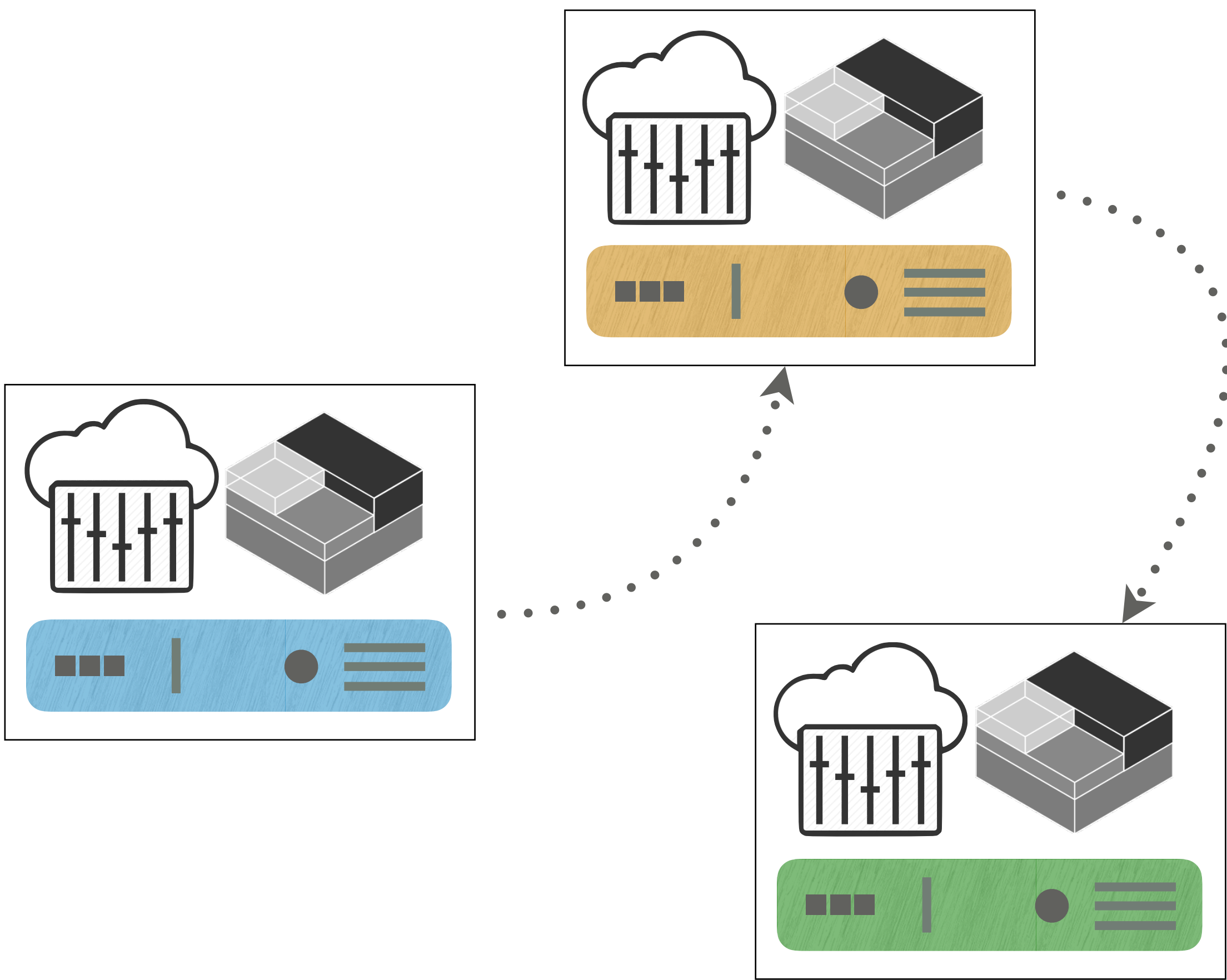


- Automaton Controller**
- Handles *server management*
 - Monitor server marketplace, server usage
 - Mange migrations and handoff



- Linux Container**
- Handles *server usage*
 - Consistent view of underlying h/w
 - Efficient system-level migration

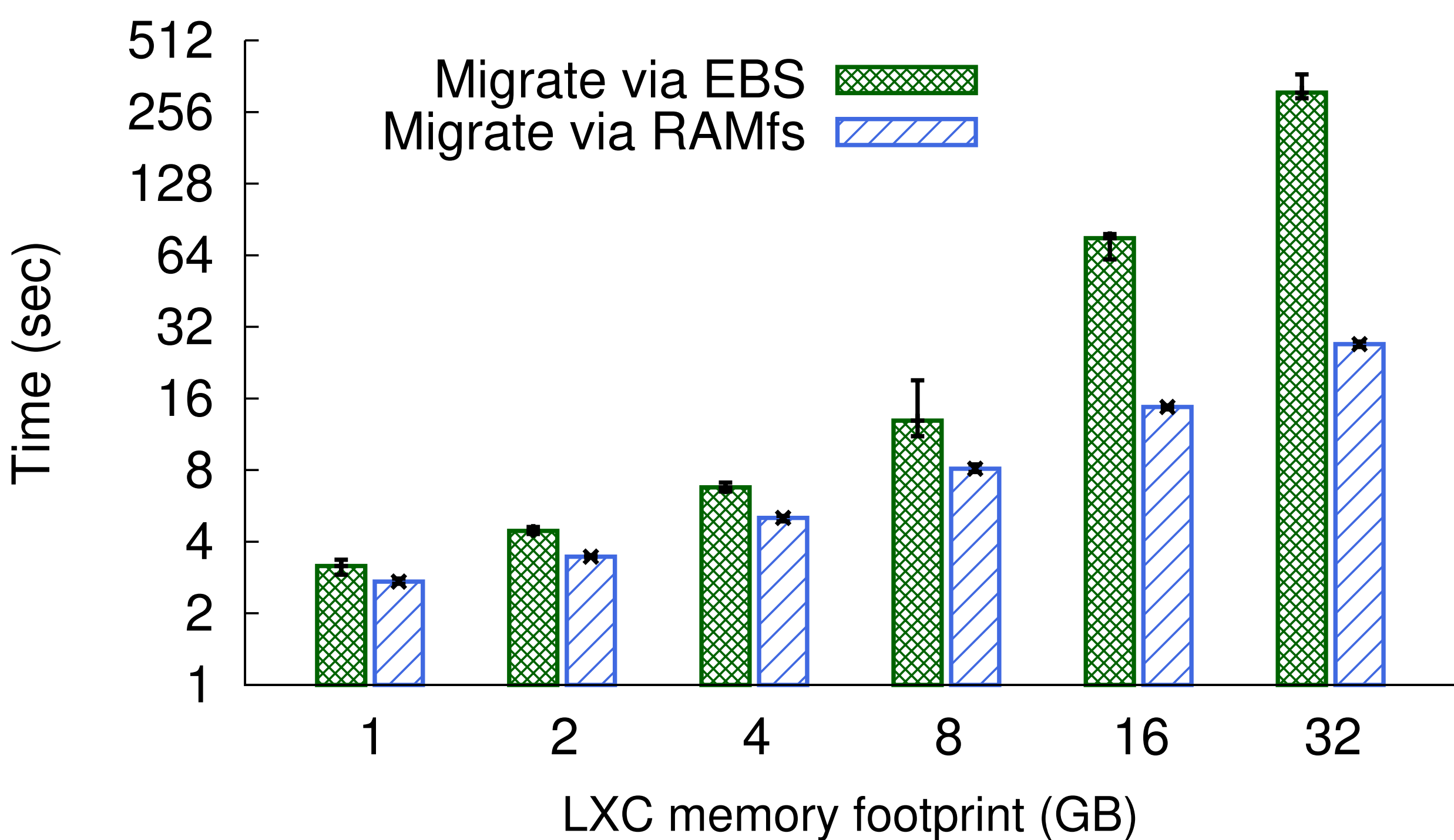
Transparent Automated Self-migration



PRELIMINARY RESULTS



Prototype on Amazon EC2
Applications <16GB in WSS could be transparently migrated within the spot warning time (120s).



Spot Market	Revocation rate (per day)
c3.2xlarge (VPC)	22.6
g2.2xlarge (VPC)	21.8
g2.8xlarge	18.0
m4.large (VPC)	17.6
m3.xlarge (VPC)	17.6
m4.xlarge (VPC)	16.4
r3.4xlarge	15.8
g2.2xlarge (VPC)	15.7
c3.2xlarge (VPC)	15.7
c3.xlarge	15.2
Automaton	7.5

Even when the choice of servers is restricted to top-10 most volatile spot markets, Automaton **reduces revocations by >50%**