SIC: Workload Anticipation for Resource and Budget Optimization in Cloud

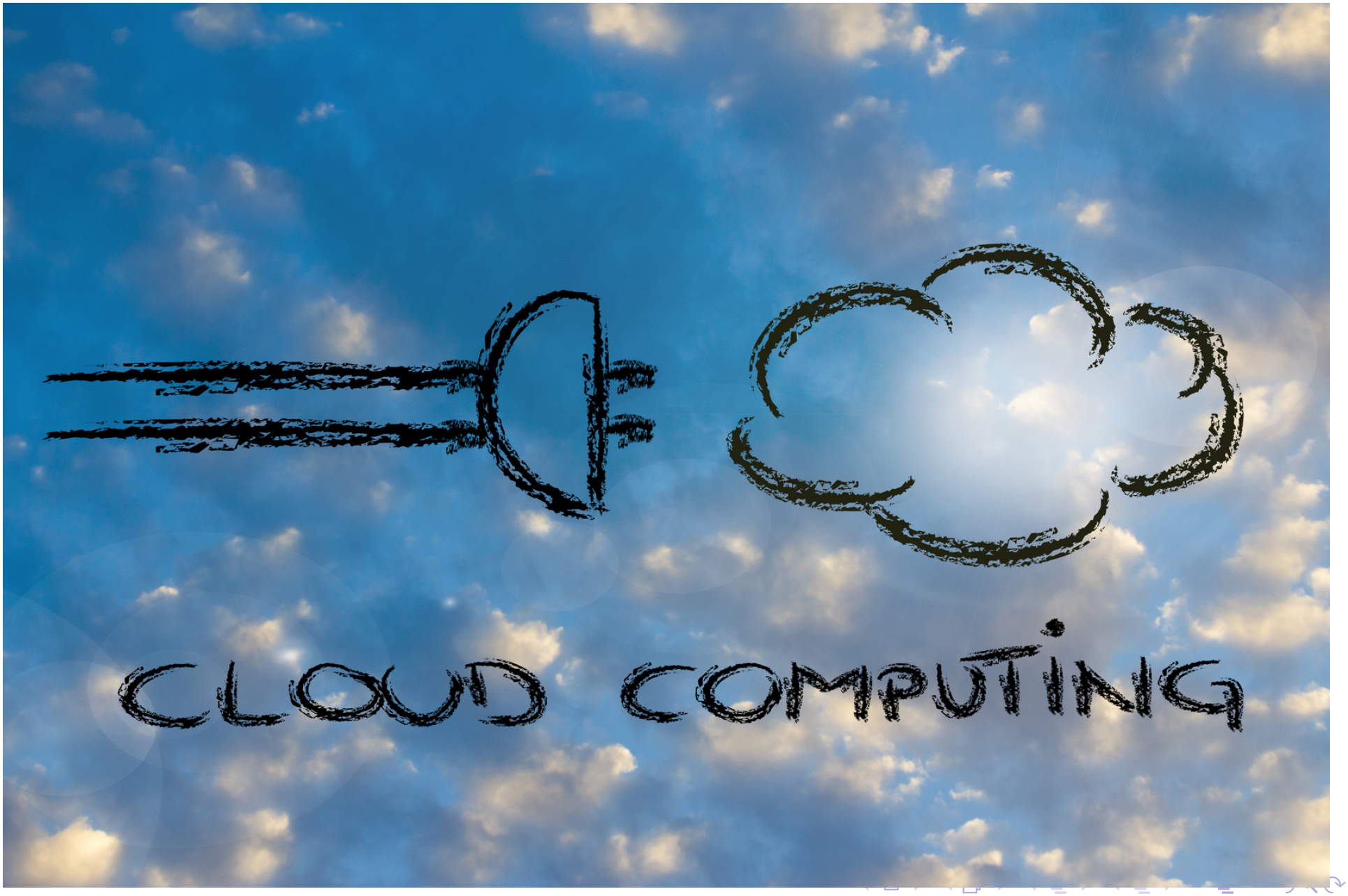
Dipendra K. Misra

Cornell University

dkm@cs.cornell.edu

3rd December 2015

~~"I wandered lonely as a cloud"~~ Rise of cloud



AI meets cloud computing

Cloud services for autonomous cars

Cloud services for *intelligent* homes

AI algorithms meet cloud computing

Requesting for an instance takes time. (1-3min on AWS)

AI algorithms meet cloud computing

Requesting for an instance takes time. (1-3min on AWS)

Keeping an instance open eats budget and resources.

AI algorithms meet cloud computing

Requesting for an instance takes time. (1-3min on AWS)

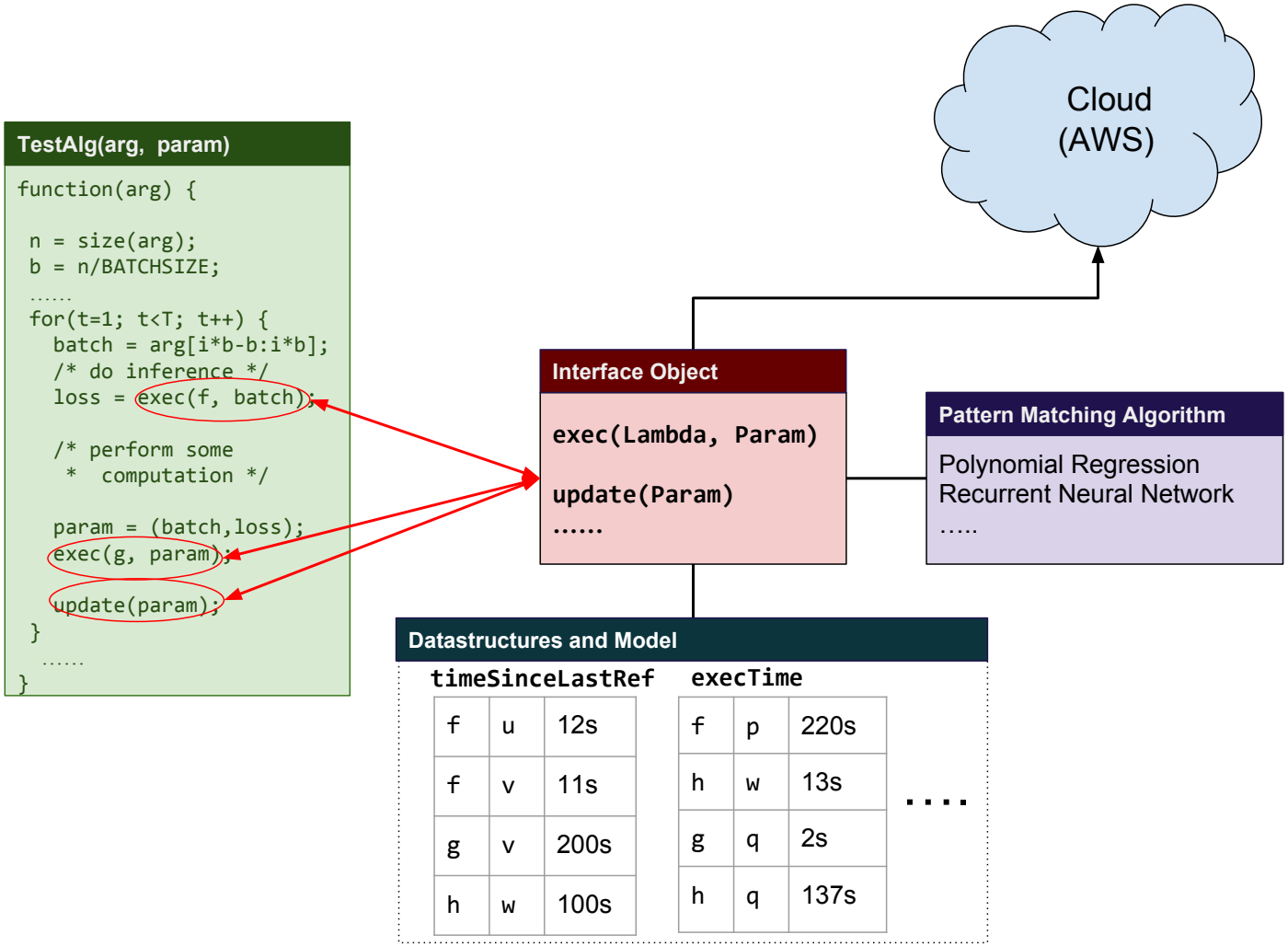
Keeping an instance open eats budget and resources.

Ideal: An instance is present when needed. (no waiting)
Only pay when using an instance. (minimal cost)

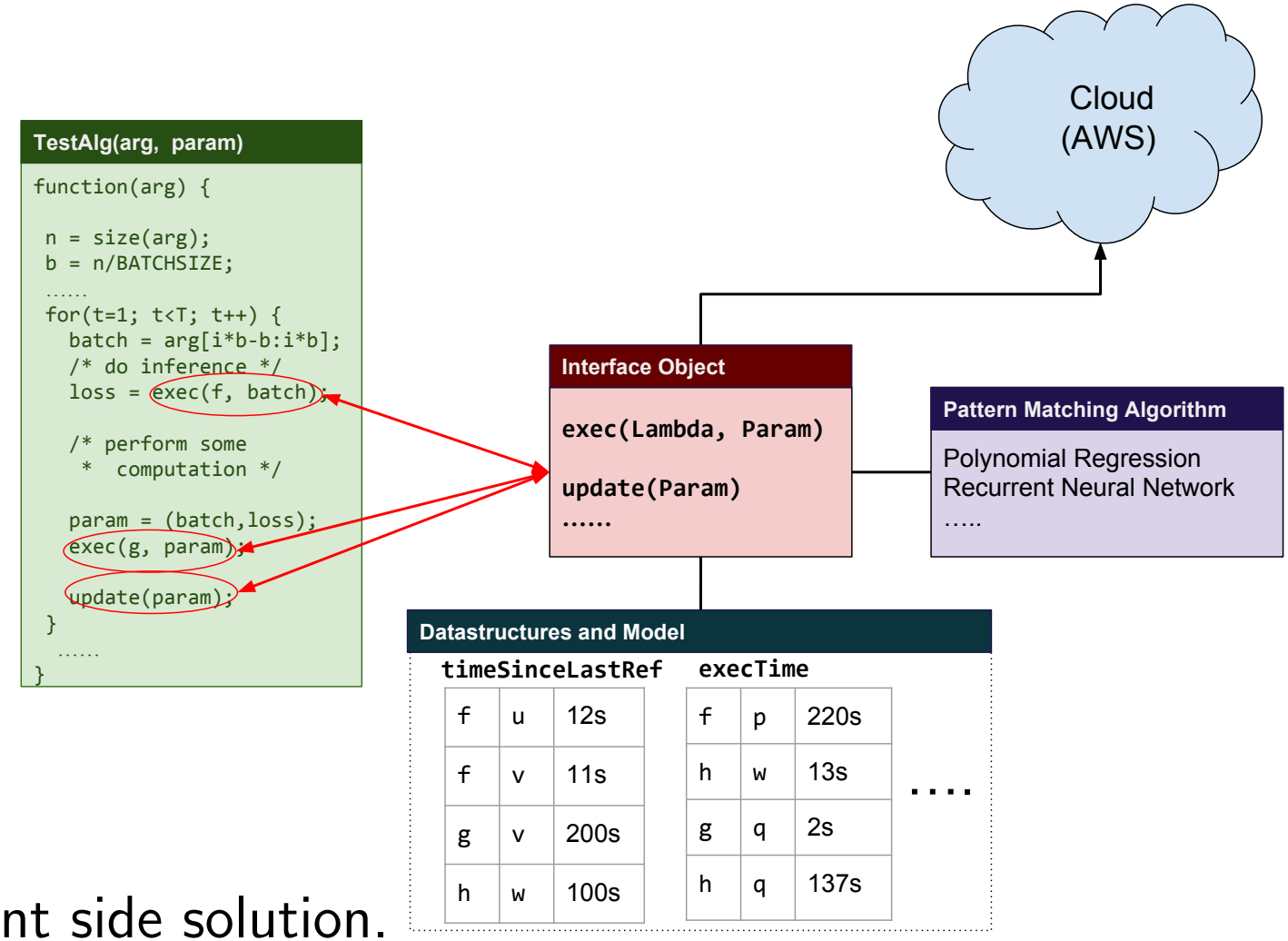
Our solution:

Predict when an instance will be needed.
Pre-launch an instance with lambda.

Our Solution: Smart Interface to Cloud (SIC)

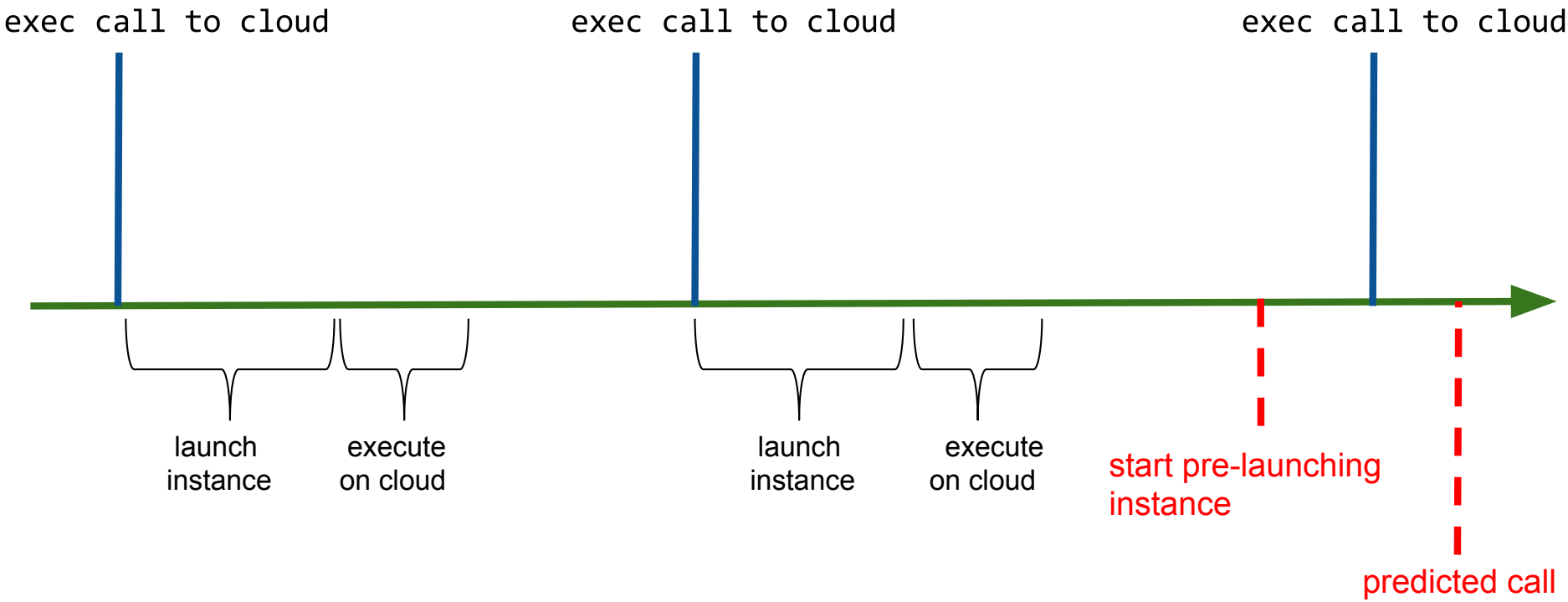


Our Solution: Smart Interface to Cloud (SIC)



- Client side solution.
- Works with arbitrary cloud services. (supports AWS)
- Oblivious to the structure of the algorithm!

SIC: Workload prediction



- Maintain statistics on call patterns, launching time for an instance etc.
- Uses **polynomial regression** to fit data and predict next call.

Results

Baselines:

- B1. Keeping an instance open all the time. (no waiting)
- B2. Launch an instance when needed and then close it. (optimal budget)

Results

Baselines:

- B1. Keeping an instance open all the time. (no waiting)
- B2. Launch an instance when needed and then close it. (optimal budget)

Algorithm	Time Taken(s)	Budget Spent*
B1	338.4s	2.68$
B2	550.4s	0.64$
SIC	517.3s	1.03$

(*cloud service rate: 1 cent per sec of use)

Thank you

Acknowledgements

Photo courtesy:

- 2nd slide cloud, http:diginomica.com
- 4th slide CMU car, www.bostonglobe.com
- 5th slide smart home, http:www.gira.com