Predictive Auto-scaling in the Kubernetes Cluster Manager

F. Matt McNaughton[1], S. Jeannie Albrecht[1], T. Brendan Burns[2]

[1]Department of Computer Science
Williams College

[2]Lead Engineer for Kubernetes
Google

Department Proposal Talk, 2016

Outline I

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work

Outline II

- Current State
- Future

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

General

Contribute to distributed system's ability to reliably and resourcefully perform large, varying amounts of computational work.

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Specific

We seek to maximize the sum of two metrics: Efficient Resource Utilization and Quality of Service.

Efficient Resource Utilization (ERU)

A measure of whether an application is efficiently using the resources it is given.

Quality of Service (QOS)

A measure of whether the application is accomplishing its stated purpose.

Balancing ERU and QOS

Our goal is to maximize the summation of ERU and QOS. We want one of the following:

- ERU to increase and QOS to stay constant.
- ERU to stay constant and QOS to increase.
- Both!

Accomplishing these goals can have substantial real world impacts.

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Cluster Managers and their Benefits

Cluster managers abstract the notion of individual computers to present multiple, network connected computers as a single chunk of computing resources.

Cluster duties include:

- Admitting/running/monitoring user submitted jobs.
- Allocating resources to jobs on the cluster.

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Overview of Cluster Managers

There are a variety of different cluster managers:

- Borg
- Mesos
- Kubernetes

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Details of Kubernetes

Cluster managers each have their own way of talking about running applications on the cluster. . . Here are the most important terms:

- Pod
- Replication Controller
- Service

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Benefits of Auto-scaling

Auto-scaling allows us to accomplish our increasing the summation of ERU and QOS.

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Overview of Auto-scaling

There are a couple of characterizations of different types of auto-scaling.

- Horizontal vs Vertical
- Reactive vs Predictive

Horizontal vs Vertical

An application being auto-scaled can have this occur through either **horizontal** or **vertical** auto-scaling.

Reactive vs Predictive

A cluster manager can determine whether to auto-scale an application based on either **reactive** or **predictive** information.

Common Types of Auto-scaling

There are three common methods of implementing auto-scaling.

- Threshold-based Rule Policies
- Time-series Analysis
- Control-theory (Feedback Control)

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Current State of Auto-scaling in Kubernetes

Kubernetes currently implements reactive, horizontal feedback control based auto-scaling.

Concerns with Auto-scaling in Kubernetes

What if it takes a long time for a new pod to be ready to handle computational work?

This thesis investigates the ability of **predictive**, horizontal feedback control auto-scaling to address the previously stated issue.

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State

Adding a predictive element allows the auto-scaling to account for the amount of time necessary for the new instance of the application to assist in sharing the work. We determine auto-scaling behavior based on the predicted future state of the application at the soonest possible time it could be ready to share work.

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. **Predictive Auto-scaling in Kubernetes**
   - Theoretical
   - **Implementation**
5. Status of Work
   - Current State

Some questions that must be answered to implement predictive, horizontal feedback control auto-scaling:

- How long does it take for a pod to be ready to share in the work?
- How can we predict the future resource utilization of an application?
- Should this behavior be enabled by default?

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Outline

1. Goals
   - General
   - Specific
2. Accomplishing General Goals: Cluster Managers and Kubernetes
   - Benefits of Cluster Managers
   - Overview of Cluster Managers
   - Kubernetes
3. Auto-scaling
   - Benefits of Auto-scaling
   - Overview of Auto-scaling
   - Current State of Auto-scaling in Kubernetes
4. Predictive Auto-scaling in Kubernetes
   - Theoretical
   - Implementation
5. Status of Work
   - Current State
   - Future

Citations I