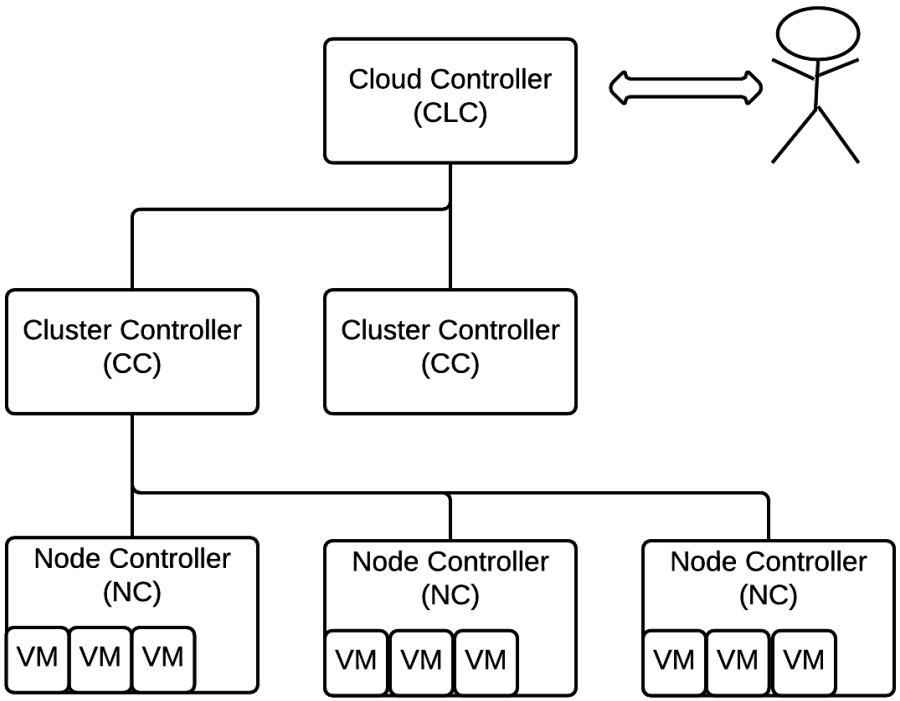
ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems

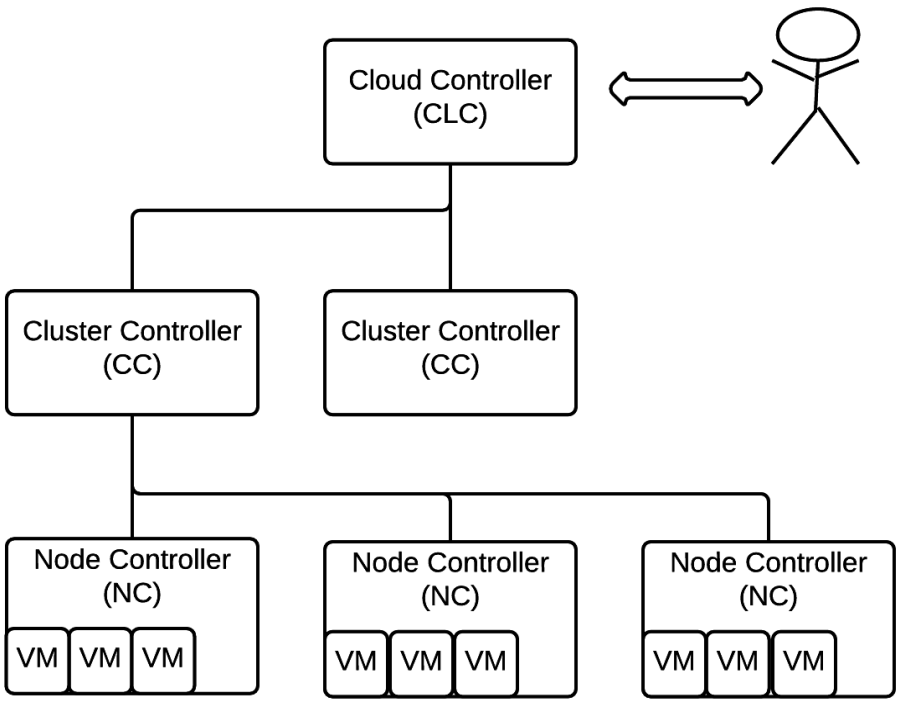
Min Du, Feifei Li

School of Computing, University of Utah

A Simplified Cloud



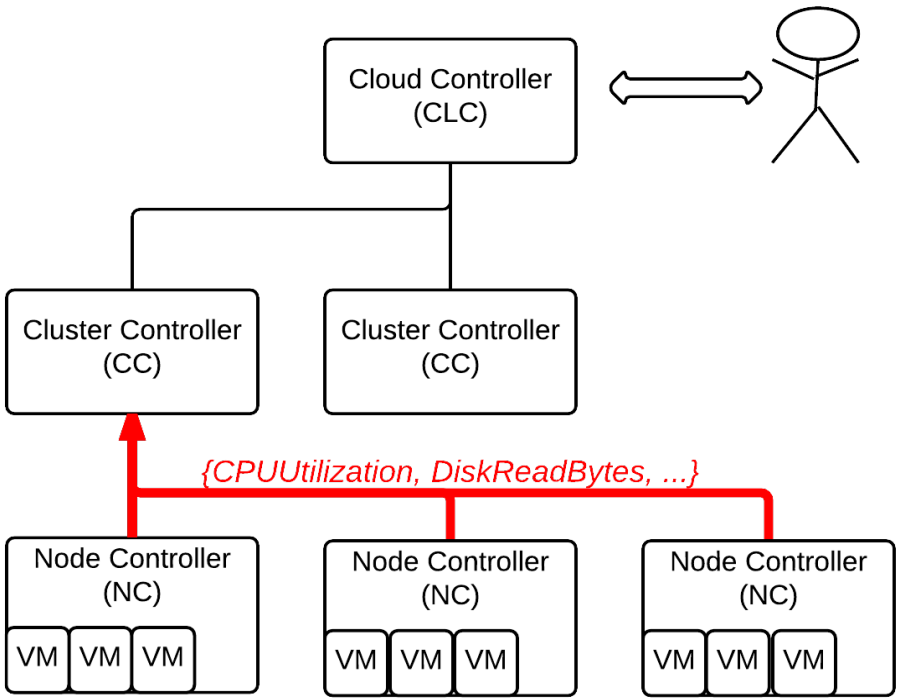
A Simplified Cloud



Monitor the Cloud

- To provide system-wide visibility
- CloudWatch (AWS/Eucalyptus)

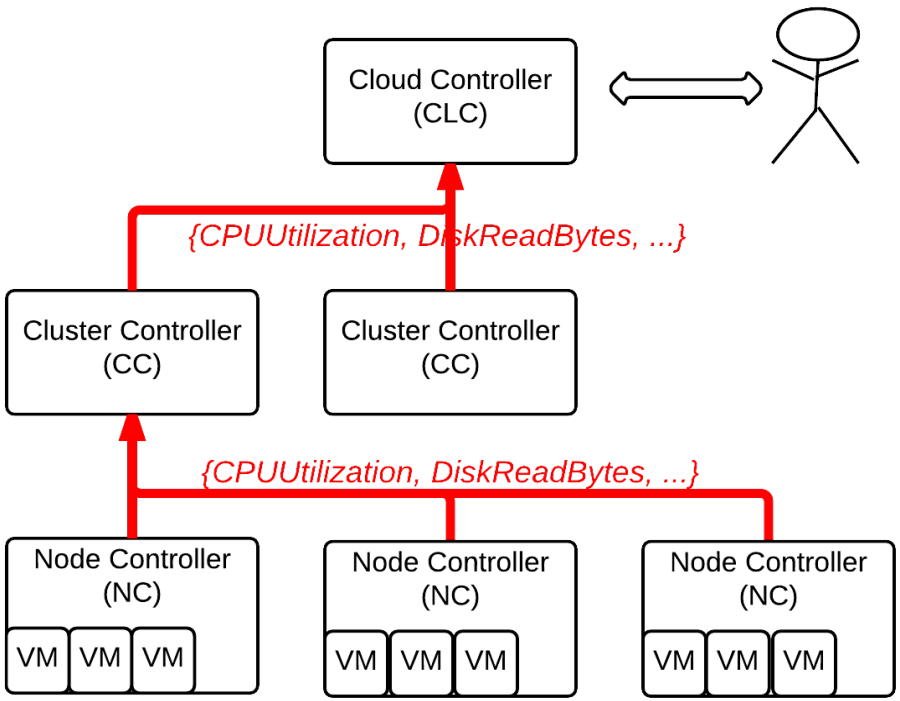
A Simplified Cloud



Monitor the Cloud

- To provide system-wide visibility
- CloudWatch (AWS/Eucalyptus)

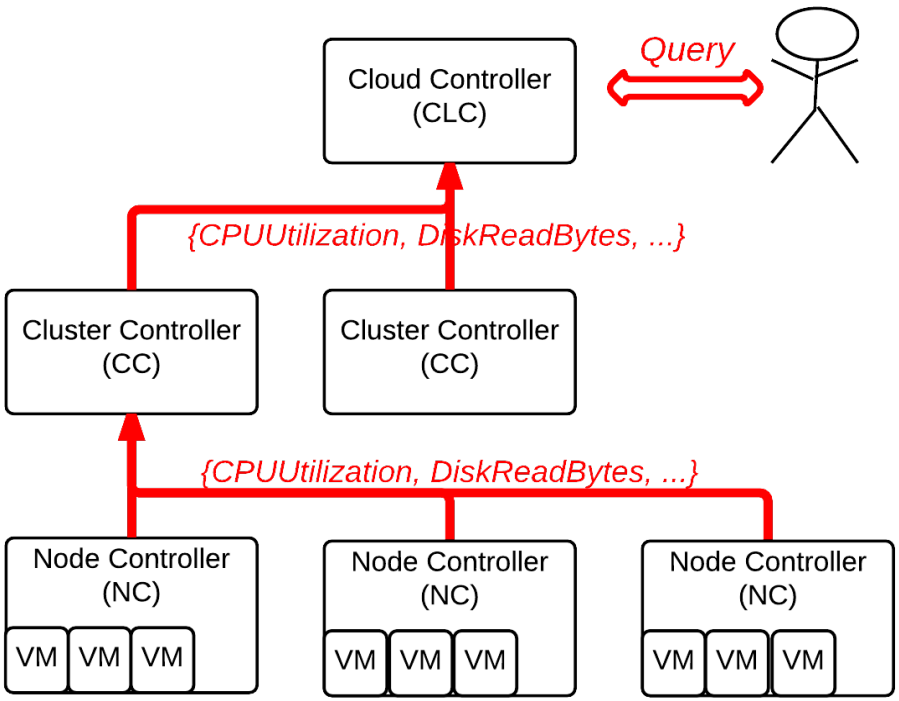
A Simplified Cloud



Monitor the Cloud

- To provide system-wide visibility
- CloudWatch (AWS/Eucalyptus)

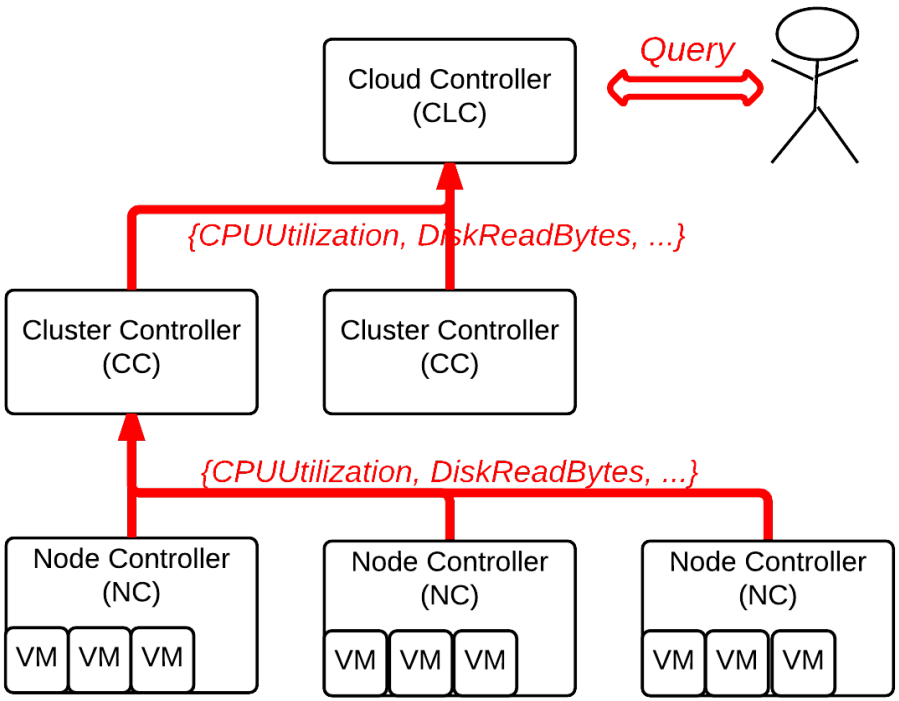
A Simplified Cloud



Monitor the Cloud

- To provide system-wide visibility
- CloudWatch (AWS/Eucalyptus)

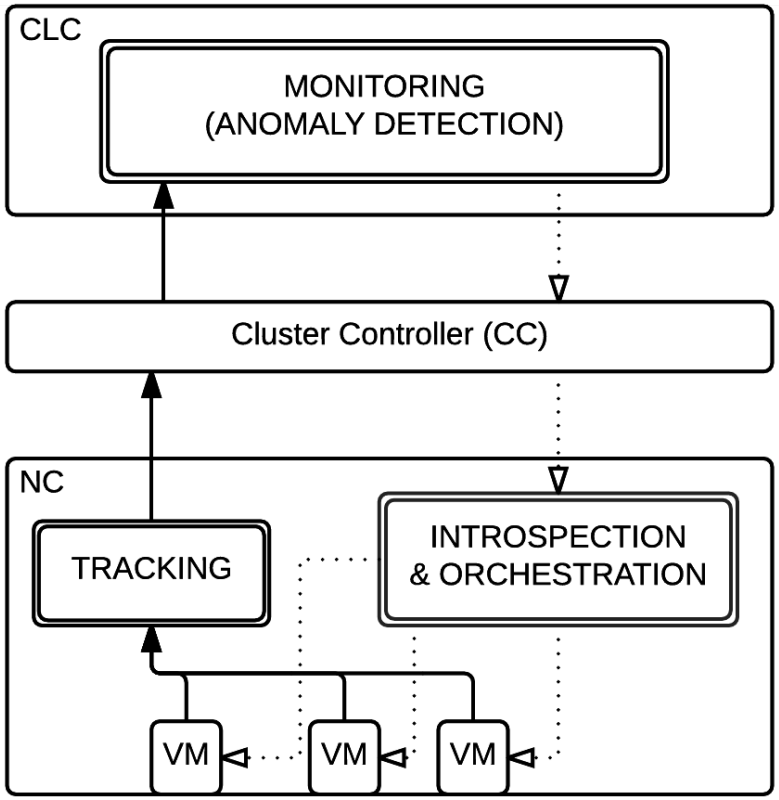
A Simplified Cloud



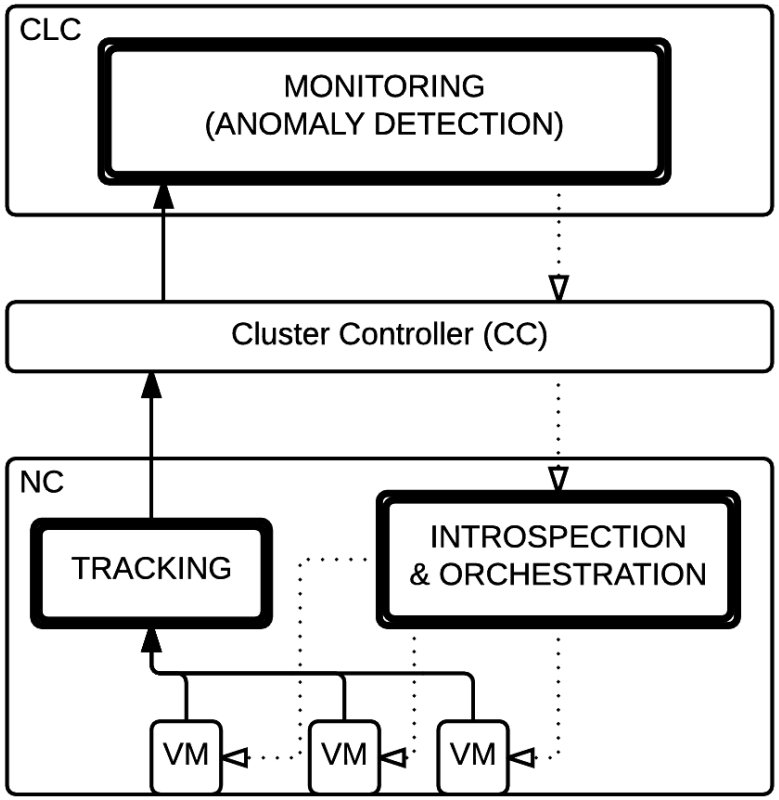
Questions

1. Monitor more efficiently?
2. Utilize the statistics for security purpose?

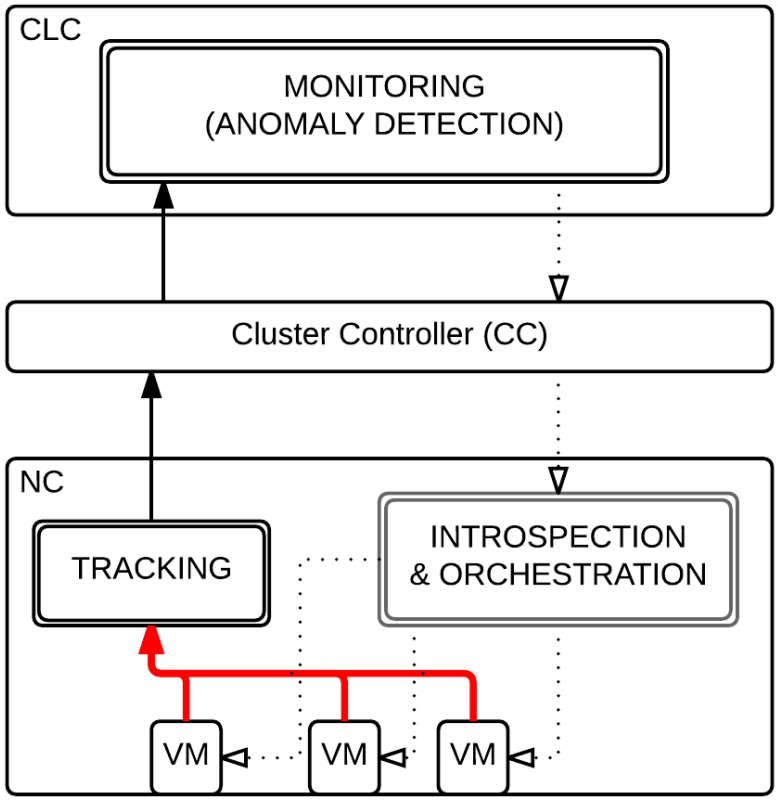
ATOM Architecture



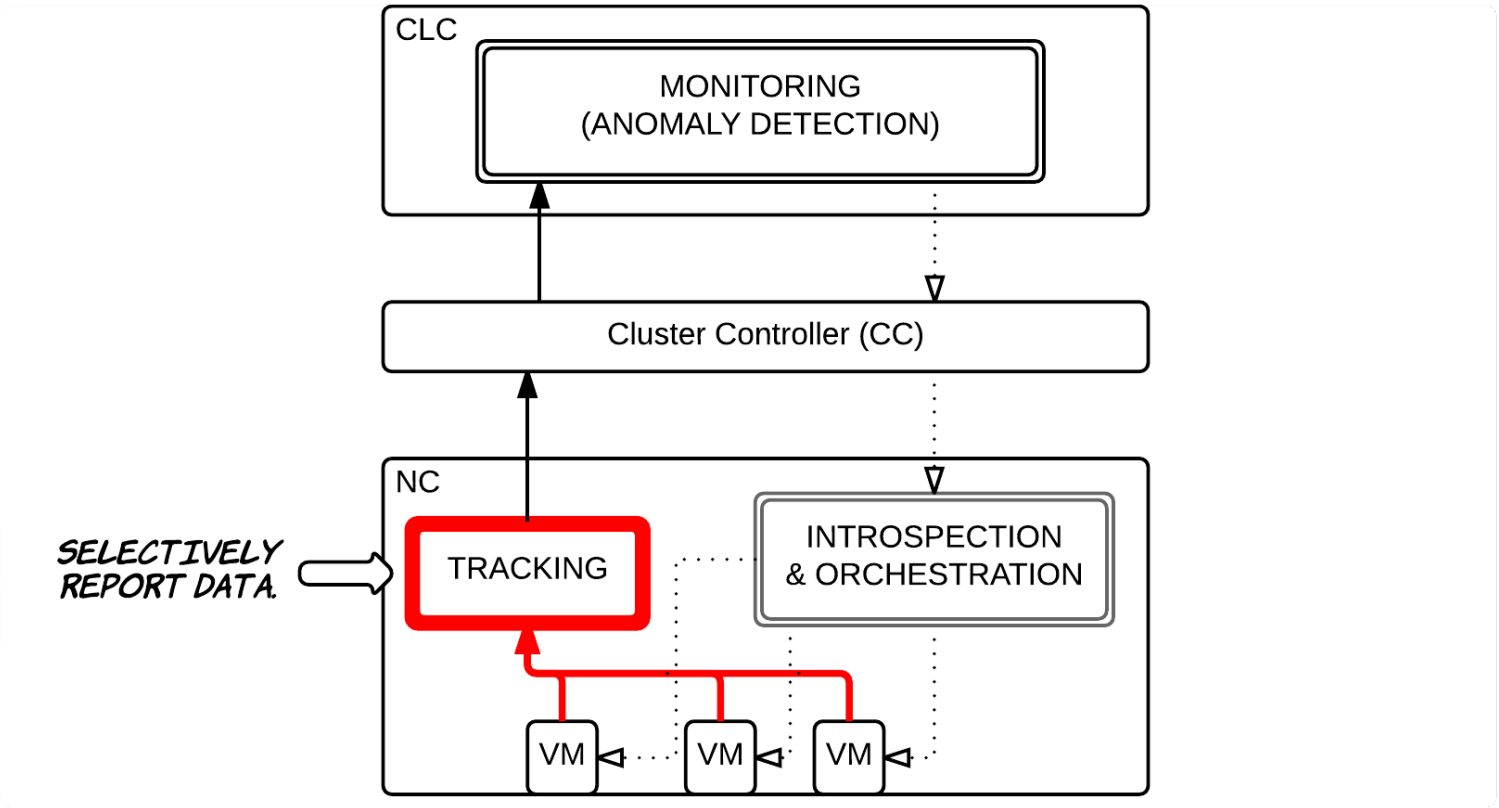
ATOM Architecture



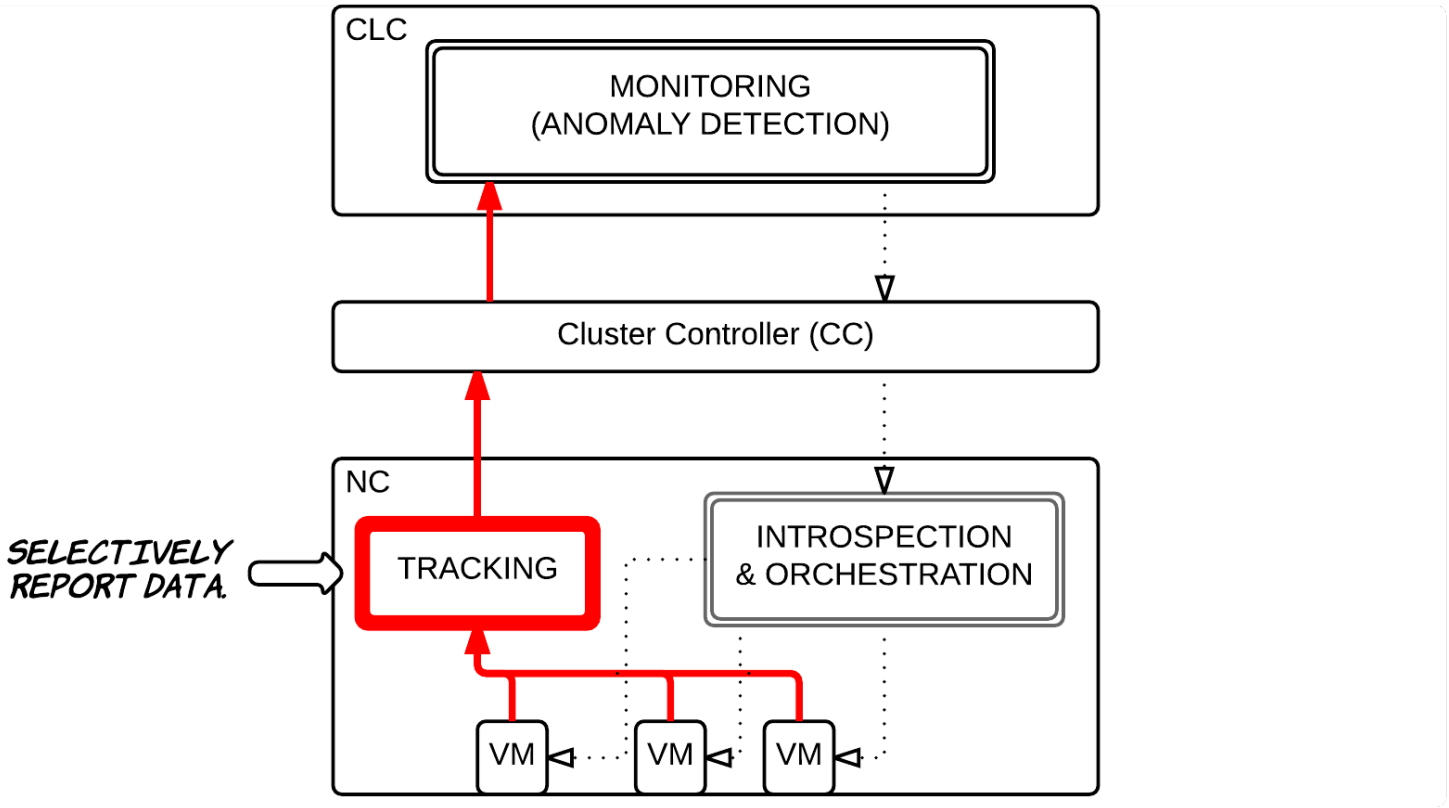
ATOM Architecture



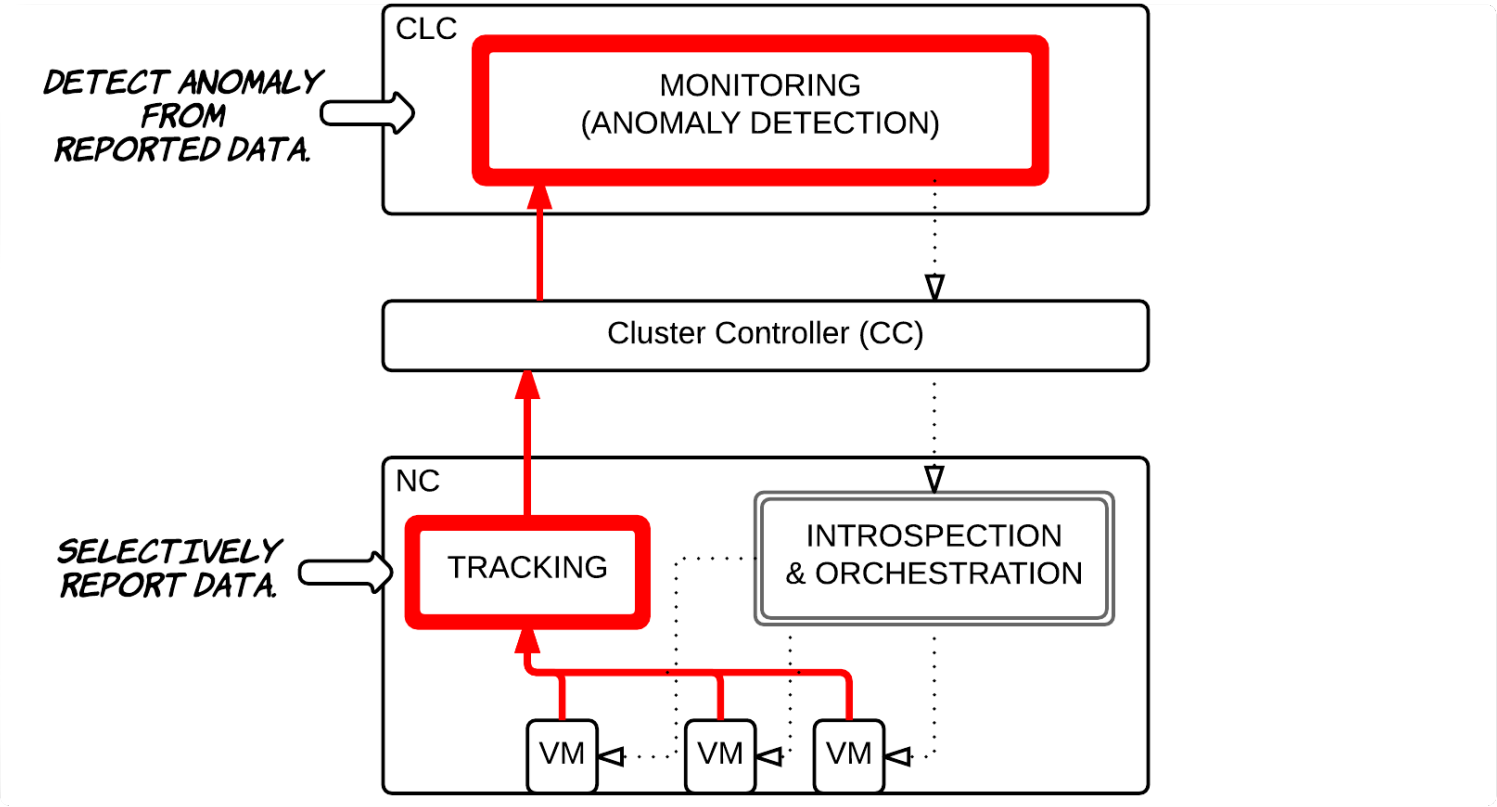
ATOM Architecture



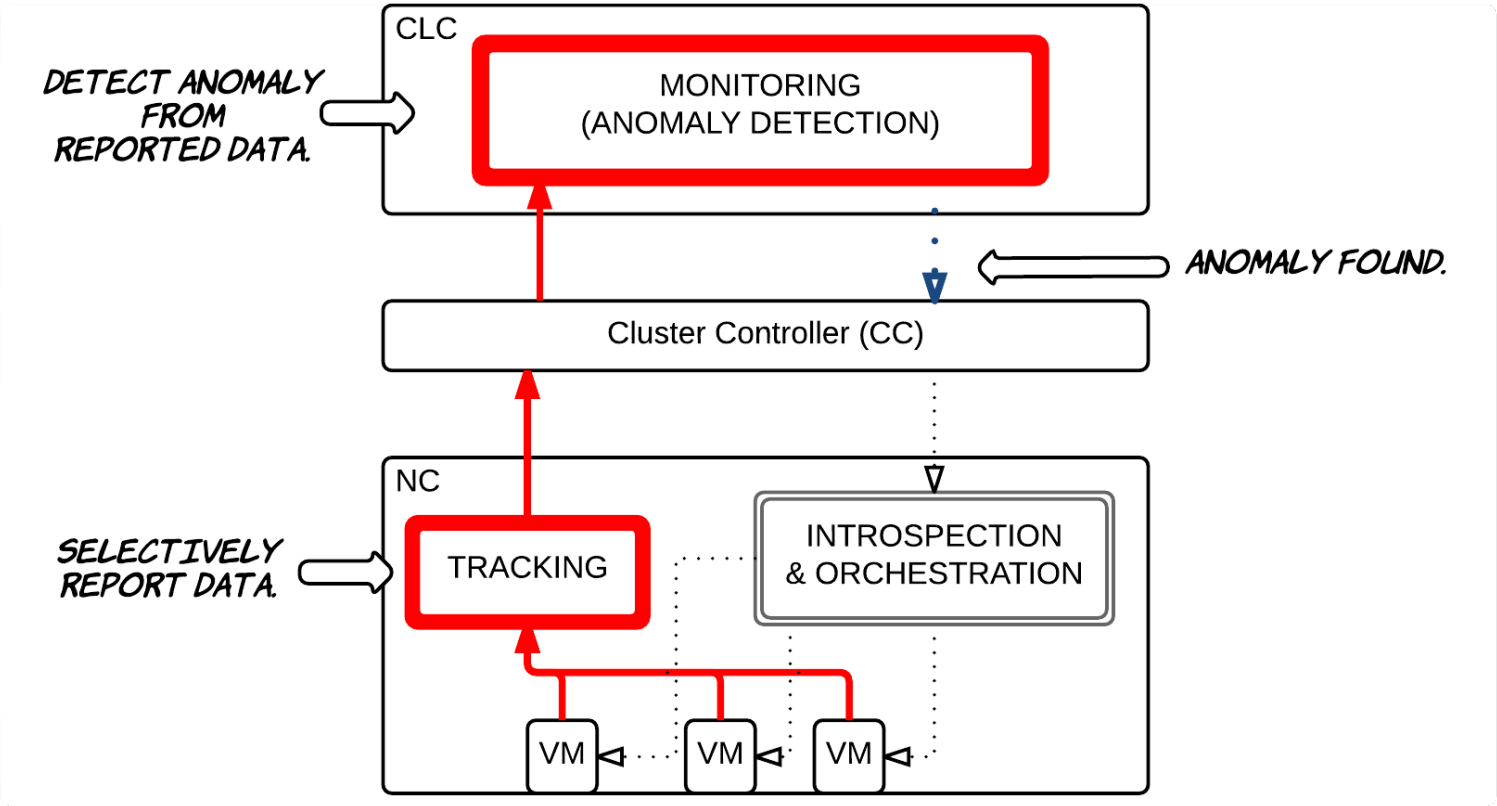
ATOM Architecture



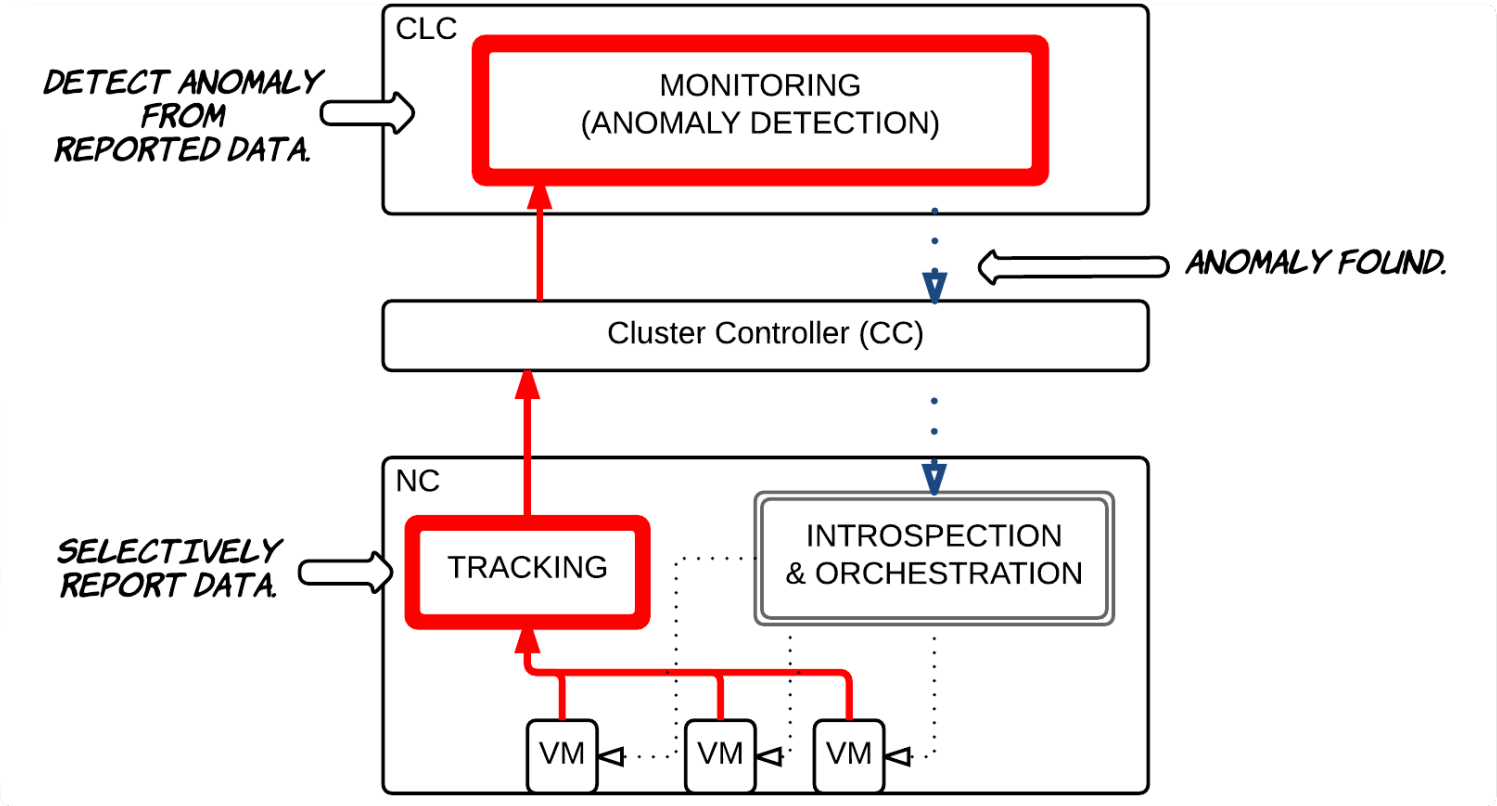
ATOM Architecture



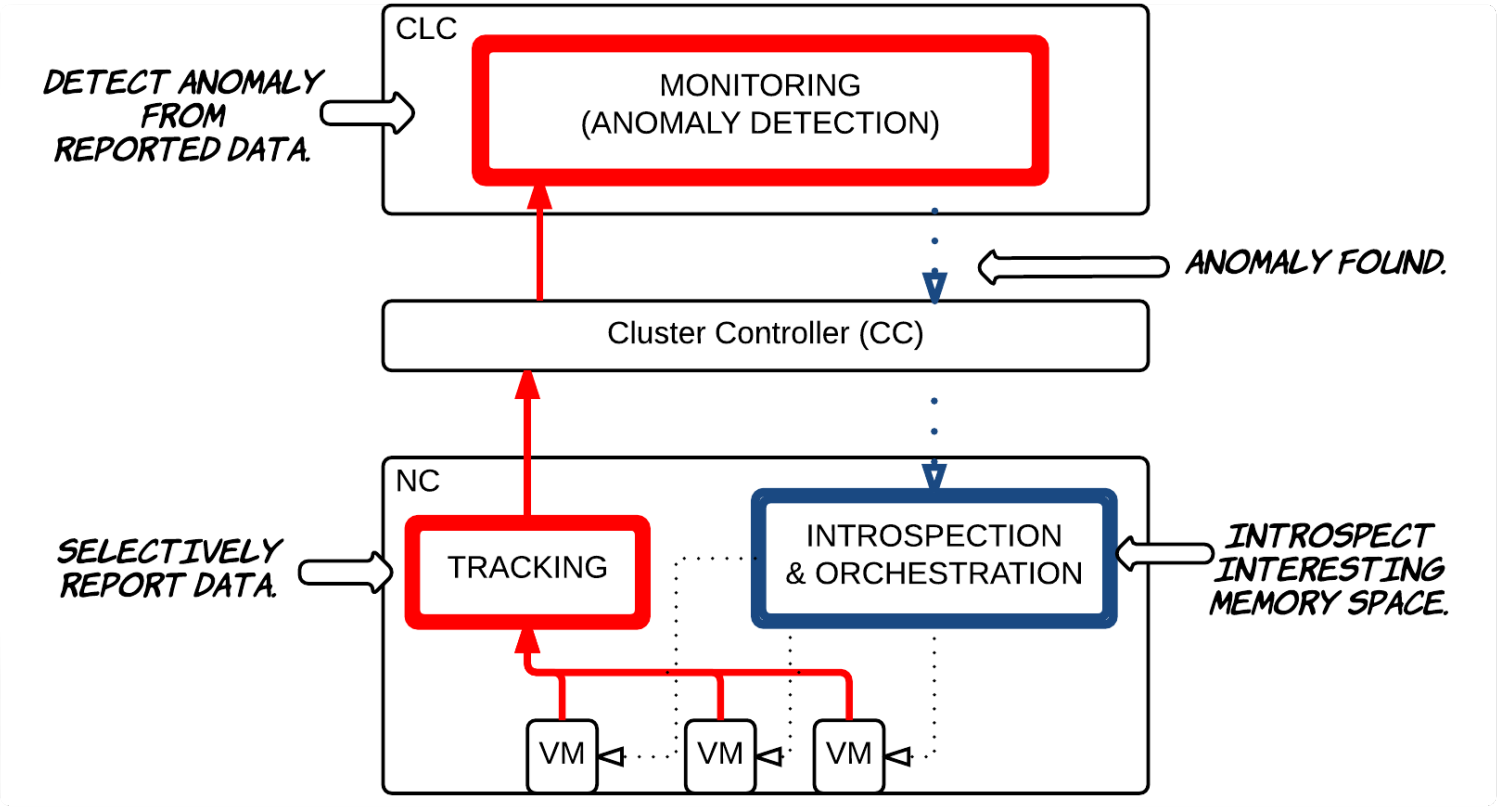
ATOM Architecture



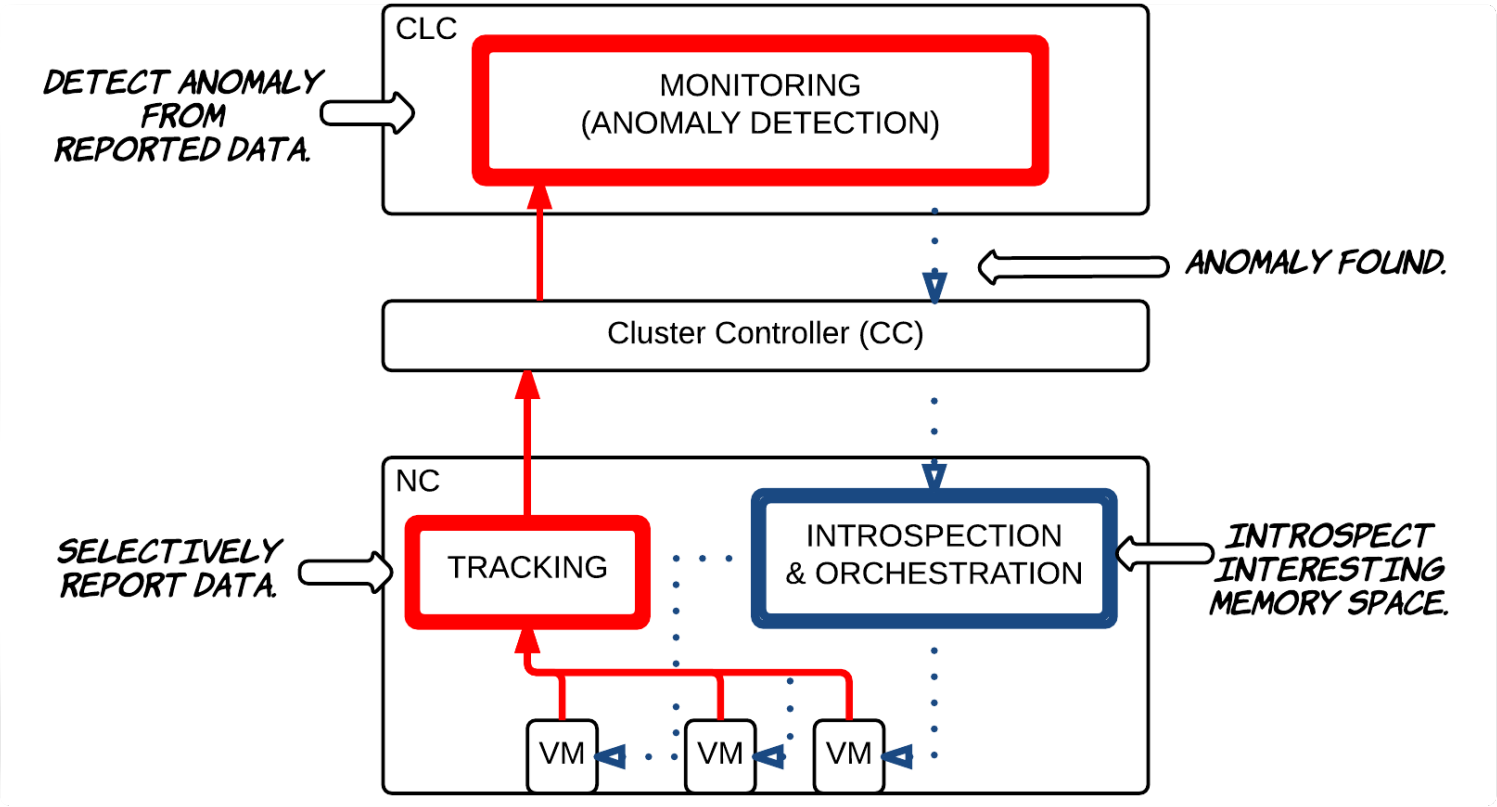
ATOM Architecture



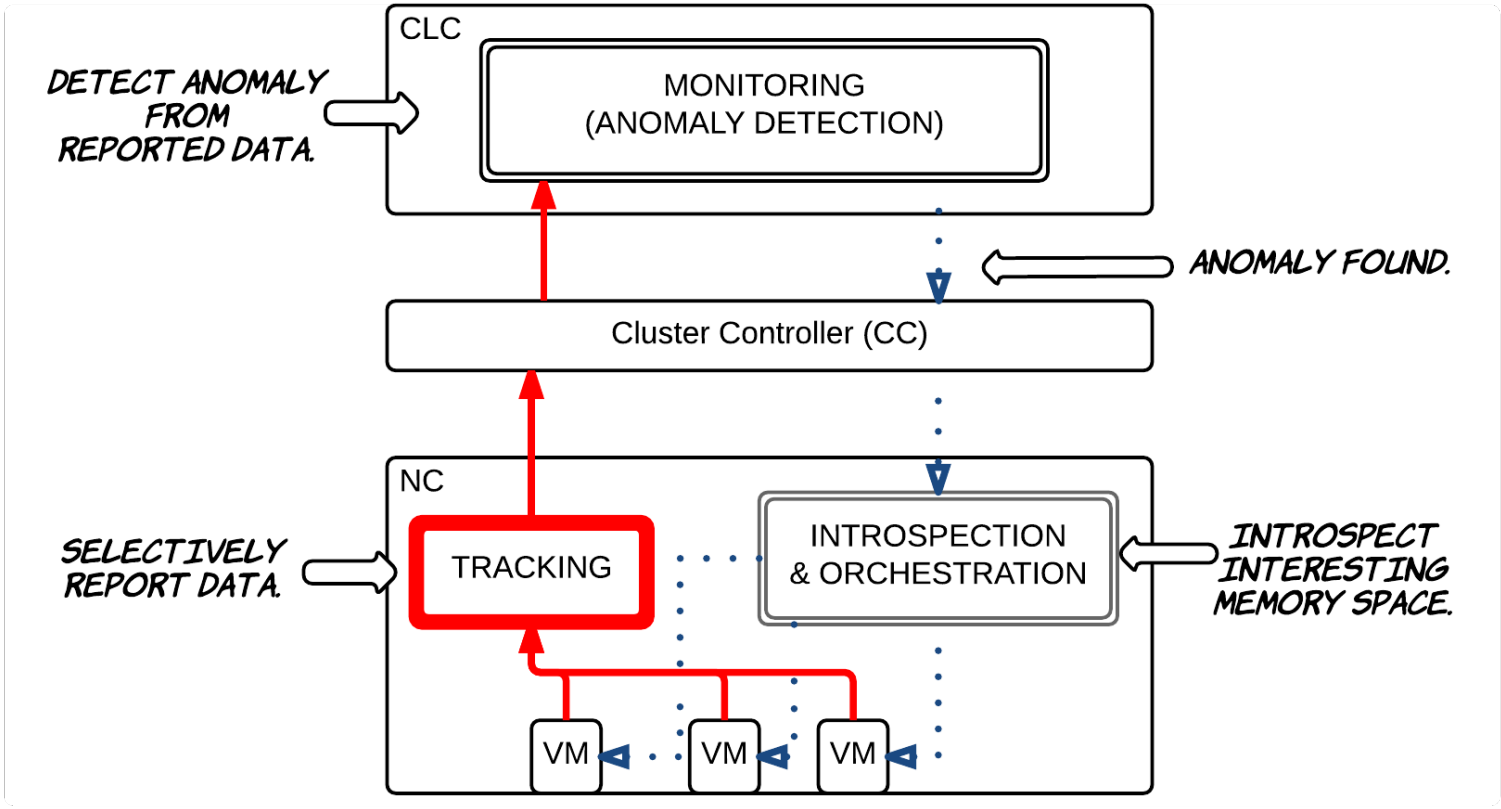
ATOM Architecture



ATOM Architecture

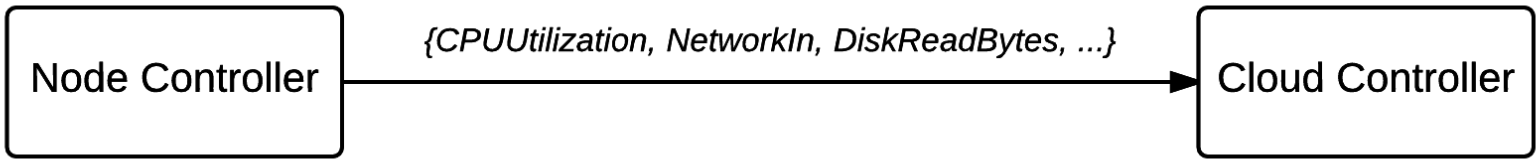


Tracking Component

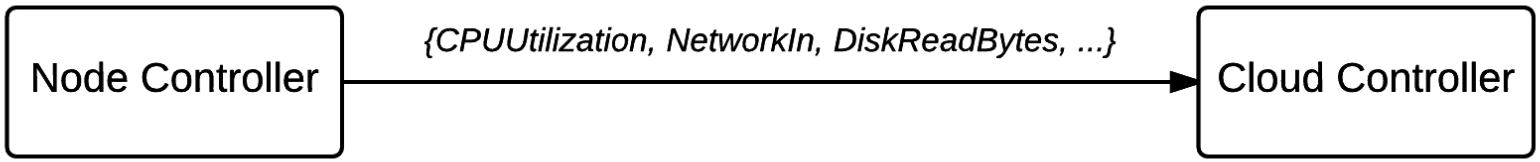


Tracking Component

Tracking Component

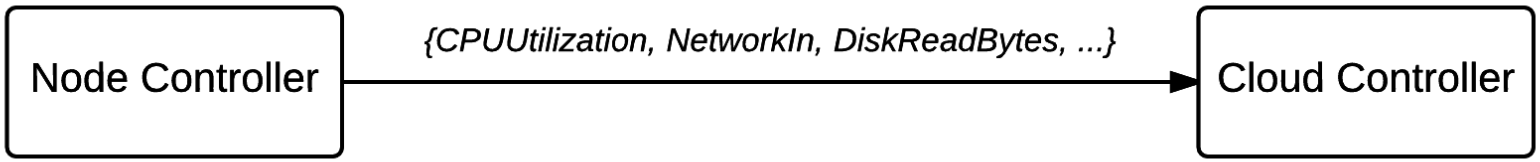


Tracking Component



What if a small error Δ is allowed?

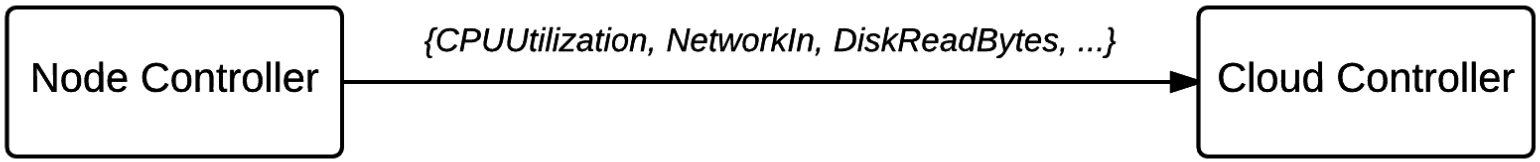
Tracking Component



What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$

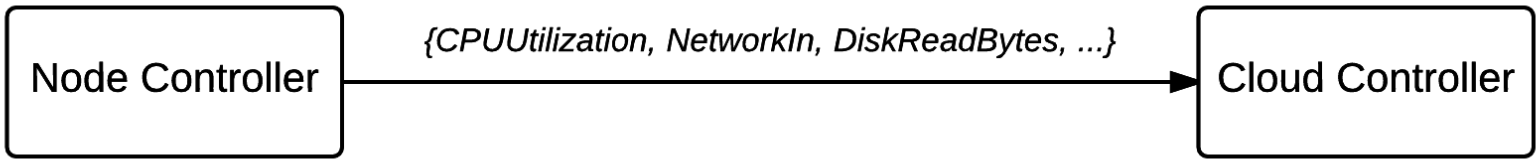
Tracking Component



What if a small error Δ is allowed?

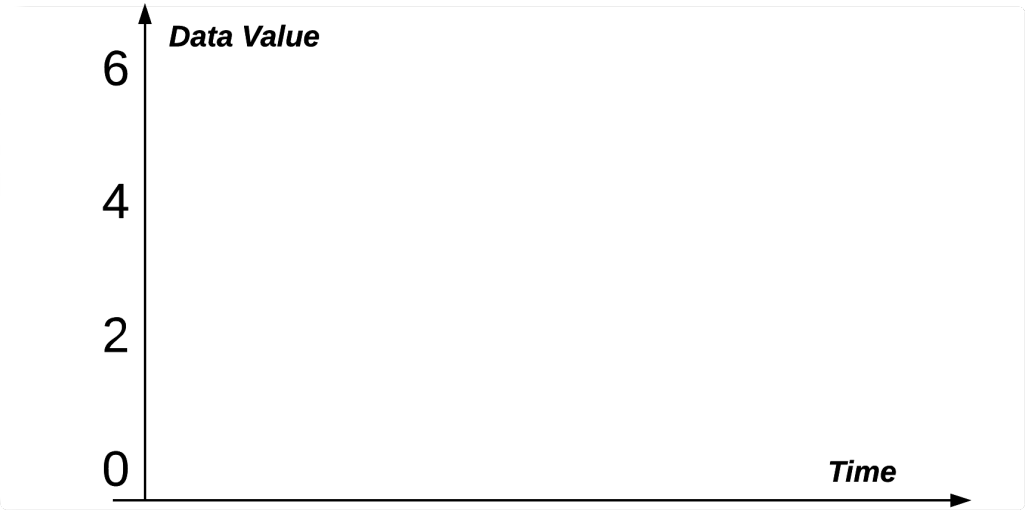
- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

Tracking Component

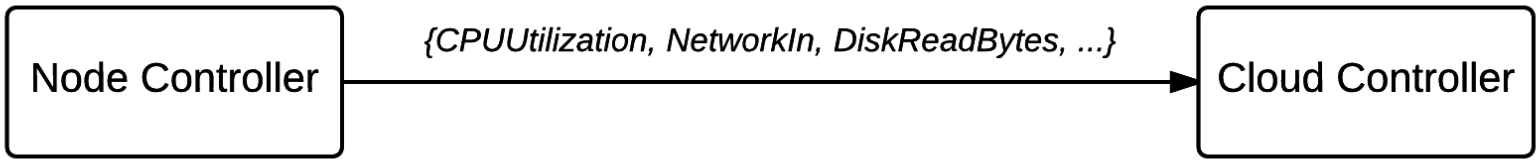


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

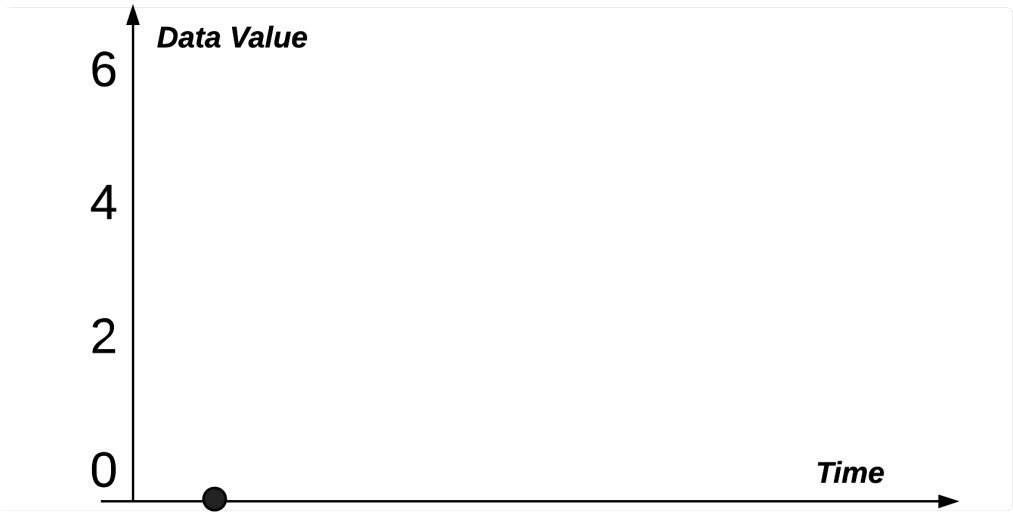


Tracking Component

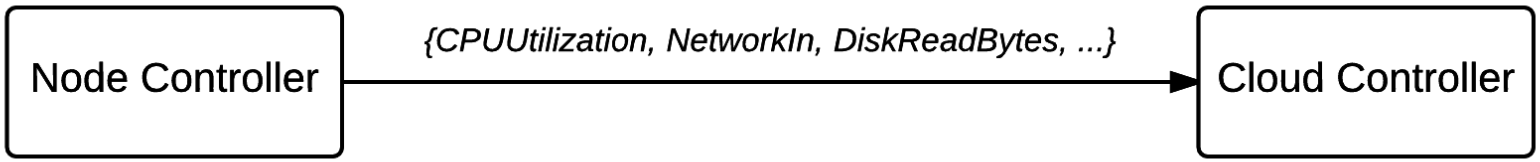


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

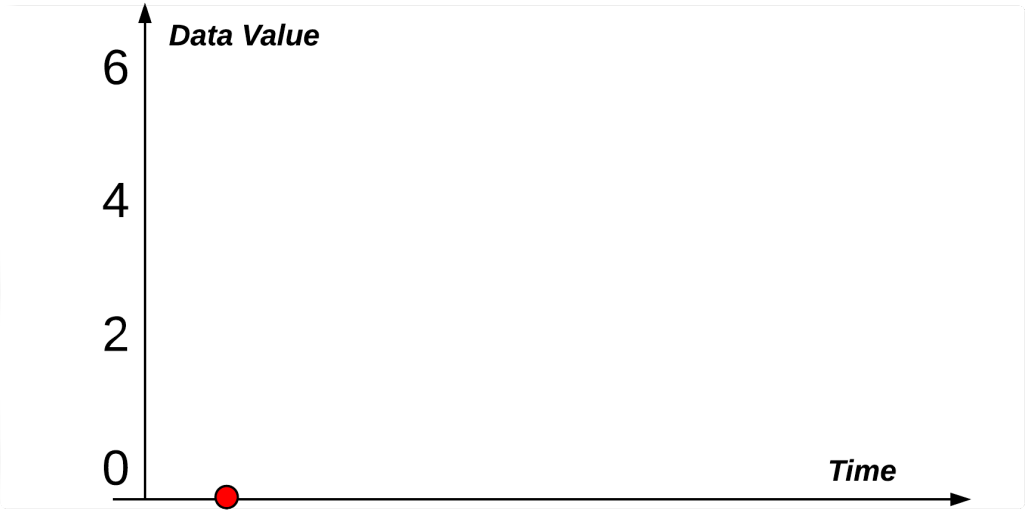


Tracking Component

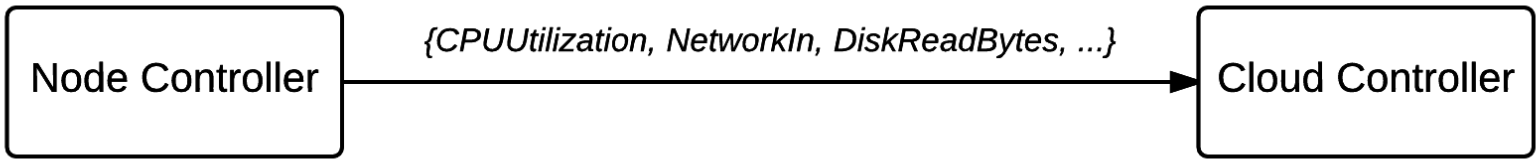


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

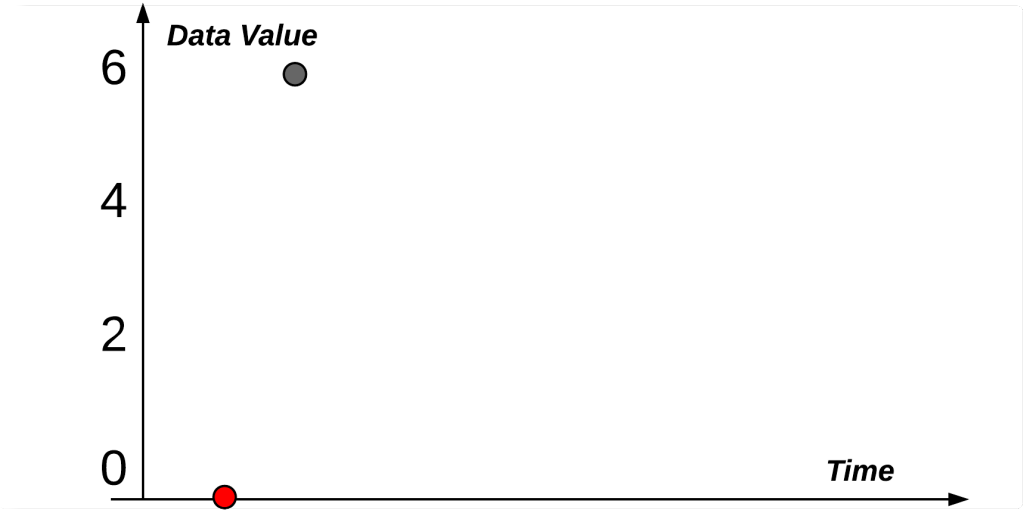


Tracking Component

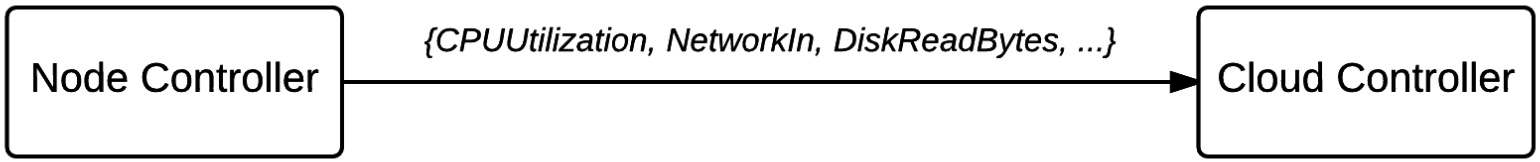


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

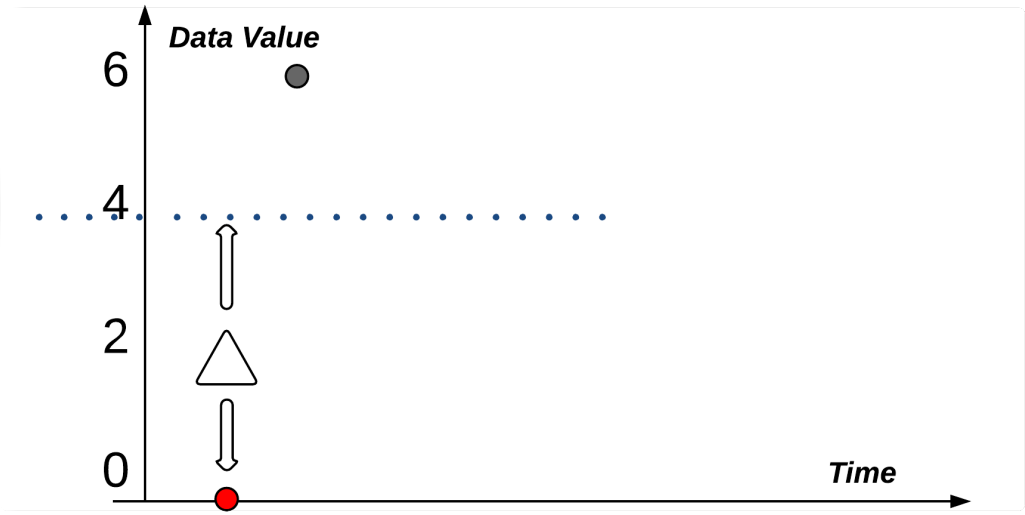


Tracking Component

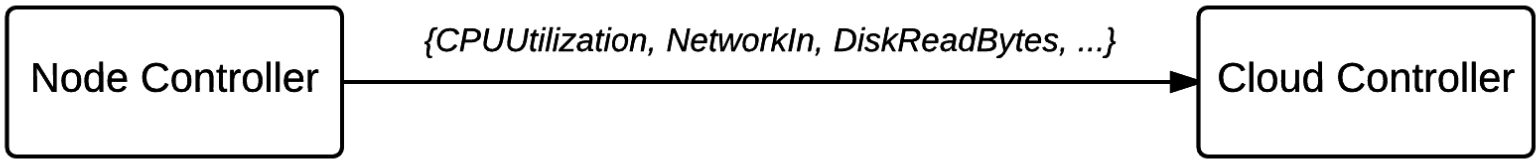


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

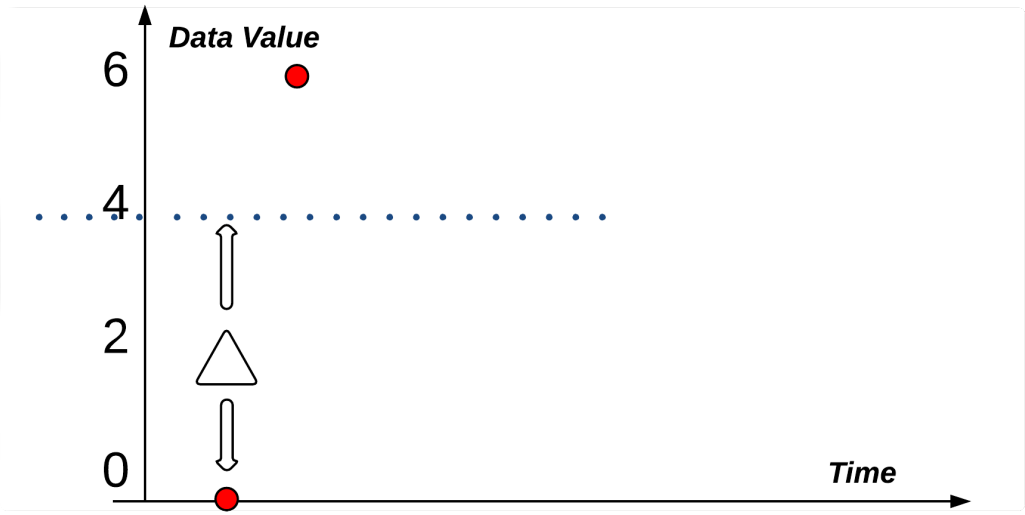


Tracking Component

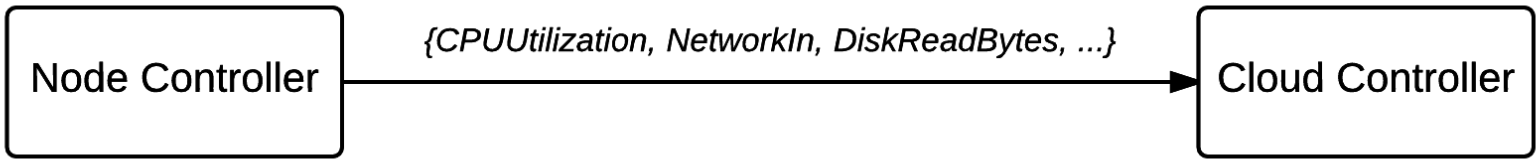


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

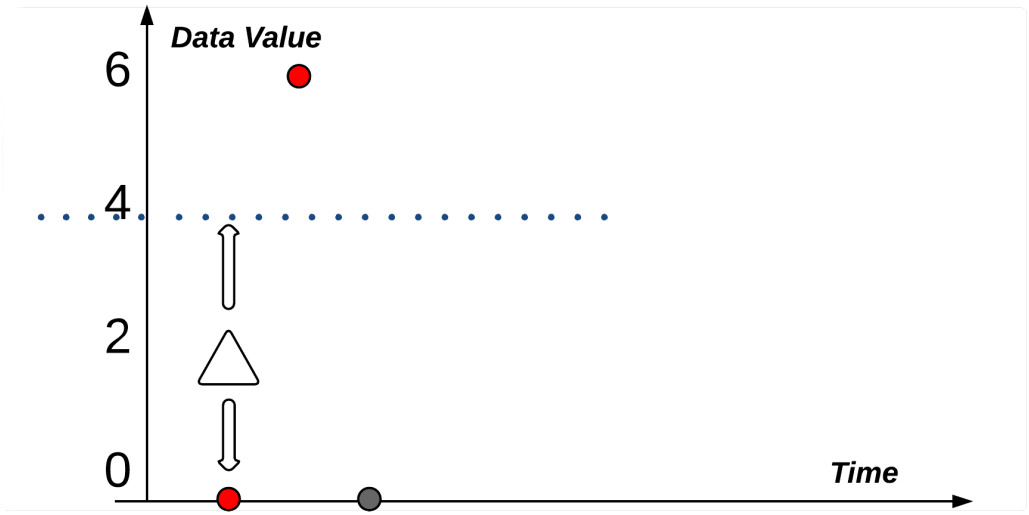


Tracking Component

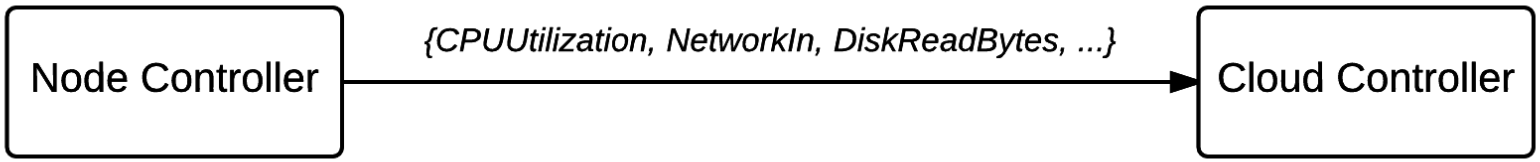


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

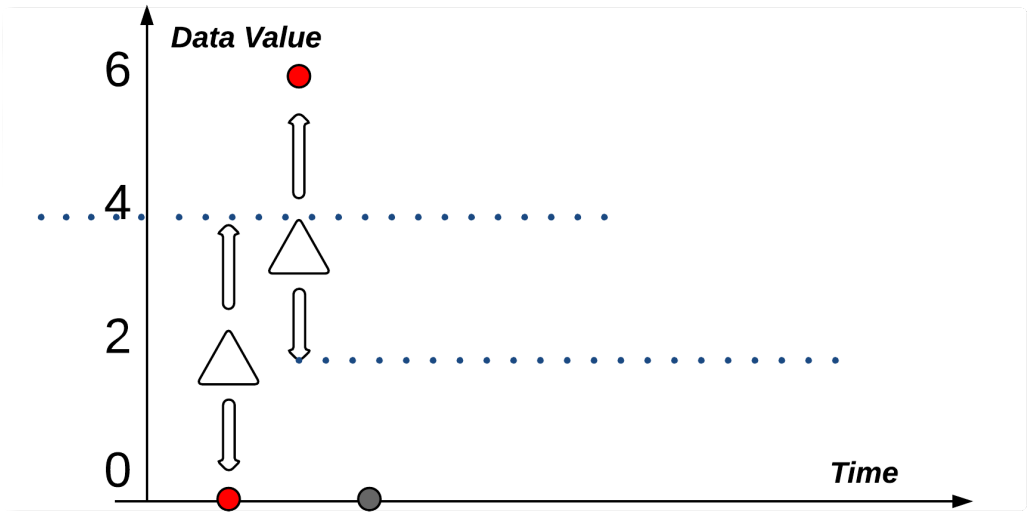


Tracking Component

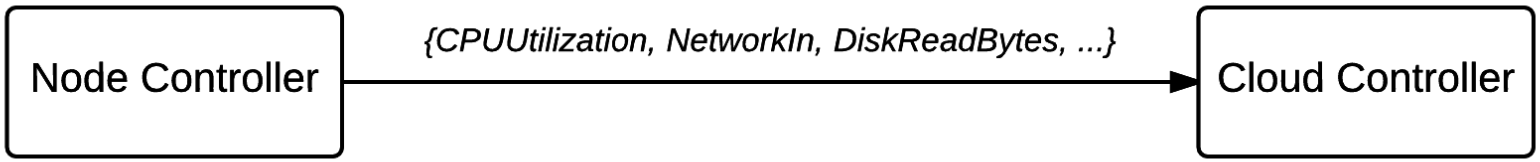


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

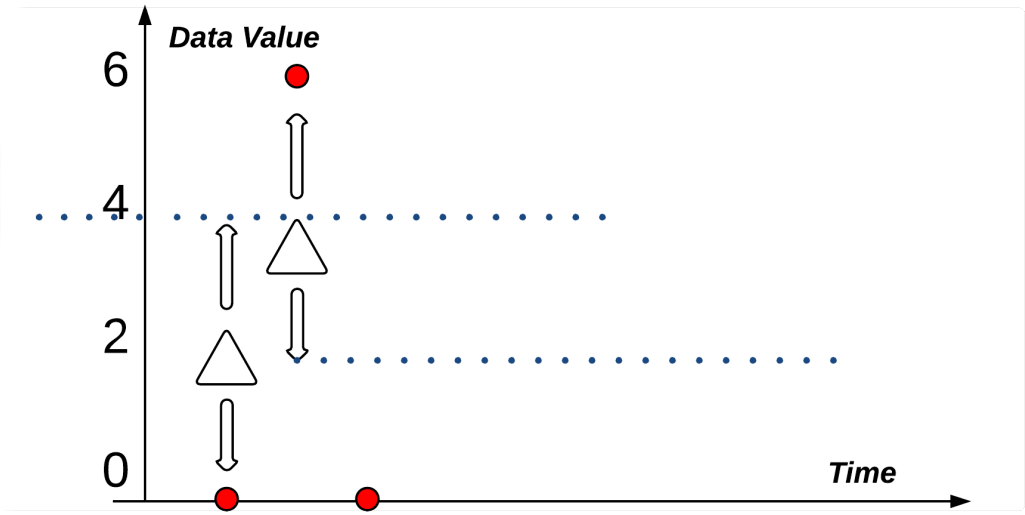


Tracking Component

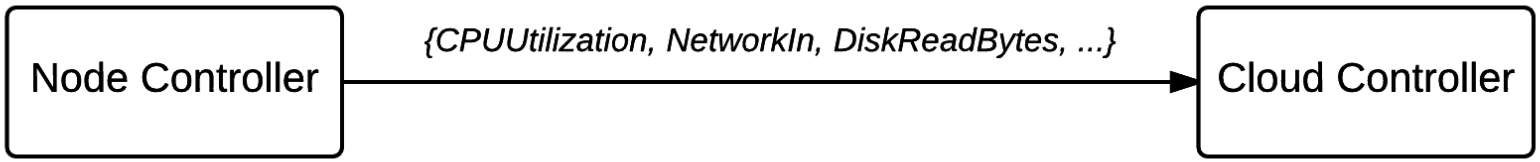


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:

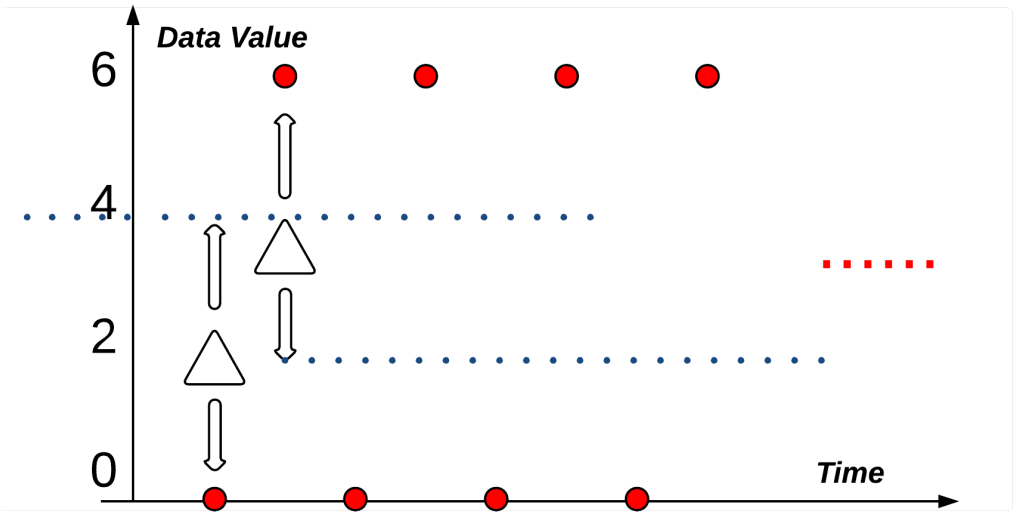


Tracking Component



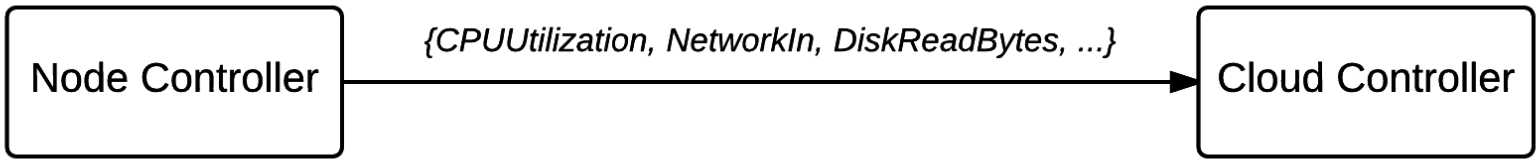
What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:



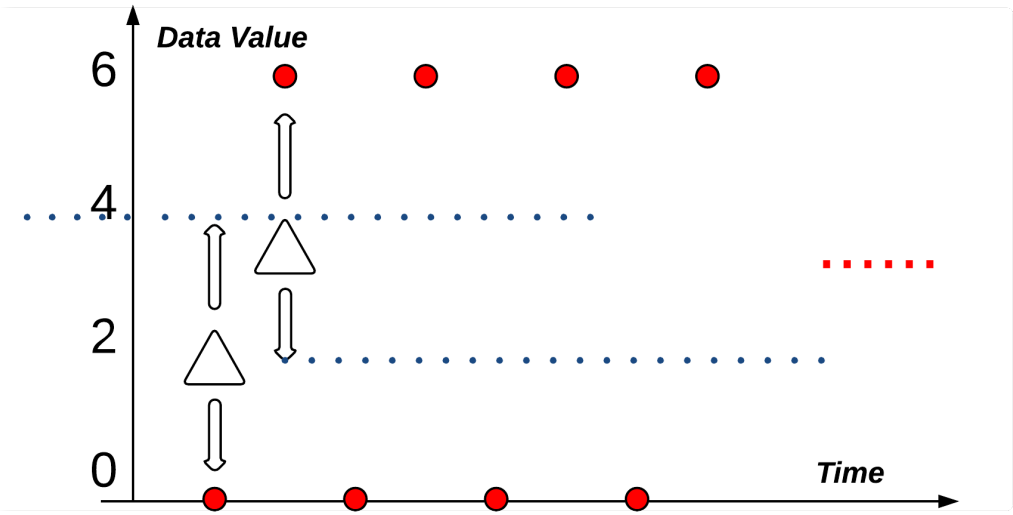
* Values sent: $\{0, 6, 0, 6, 0, 6, ...\}$

Tracking Component



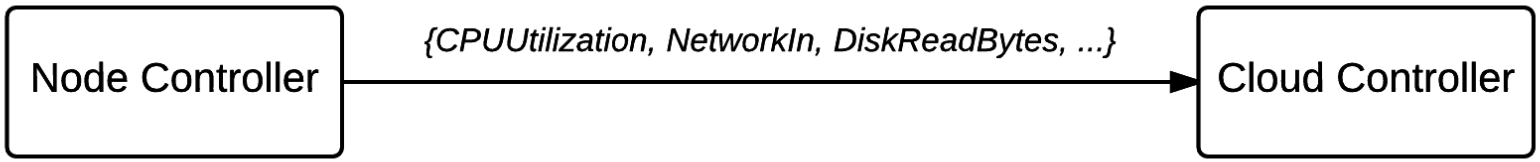
What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:



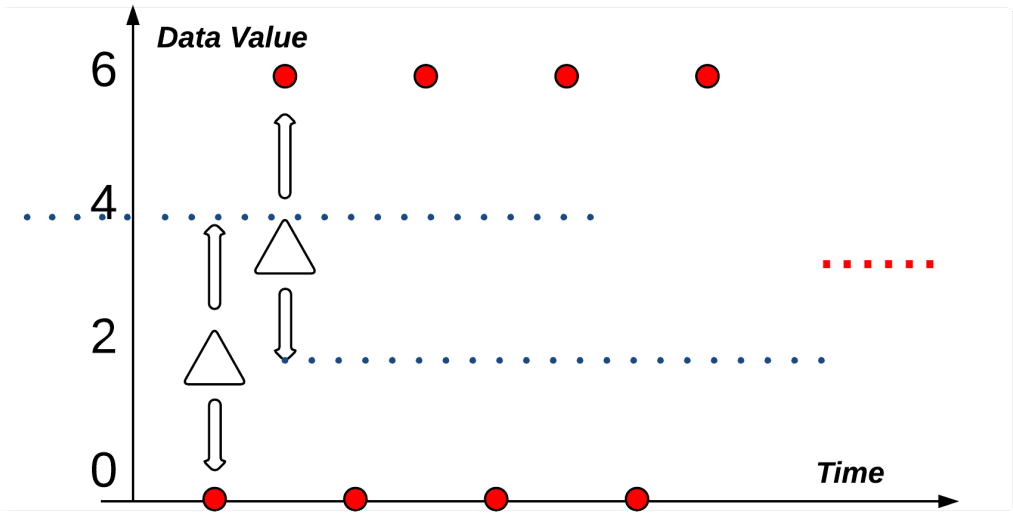
* Values sent: $\{0, 6, 0, 6, 0, 6, ...\}$
* Optimal offline algorithm could only send one value: 3

Tracking Component



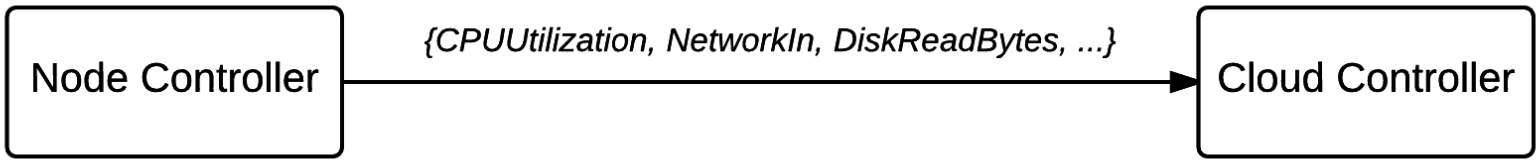
What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- A naive way:



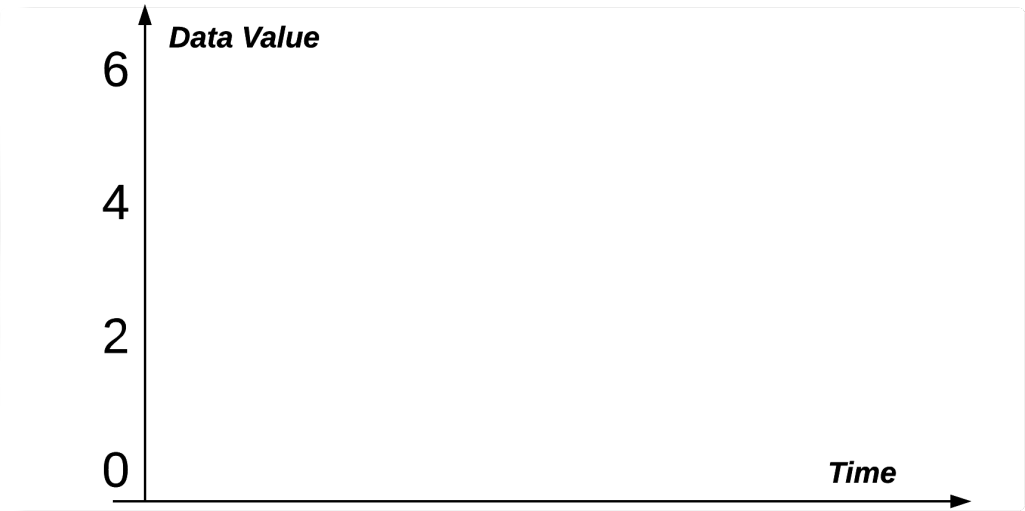
* Values sent: $\{0, 6, 0, 6, 0, 6, ...\}$
* Optimal offline algorithm could only send one value: 3
* Competitive ratio: Unbounded

Tracking Component

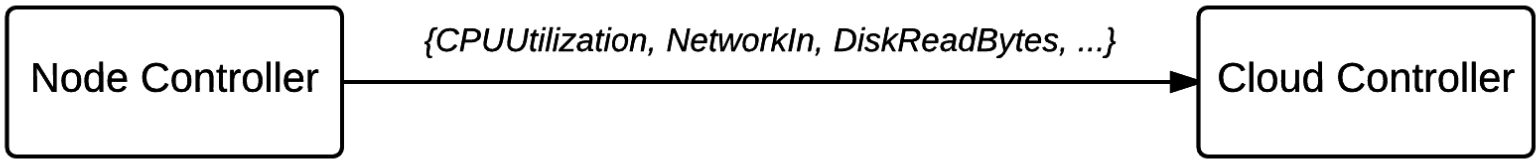


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

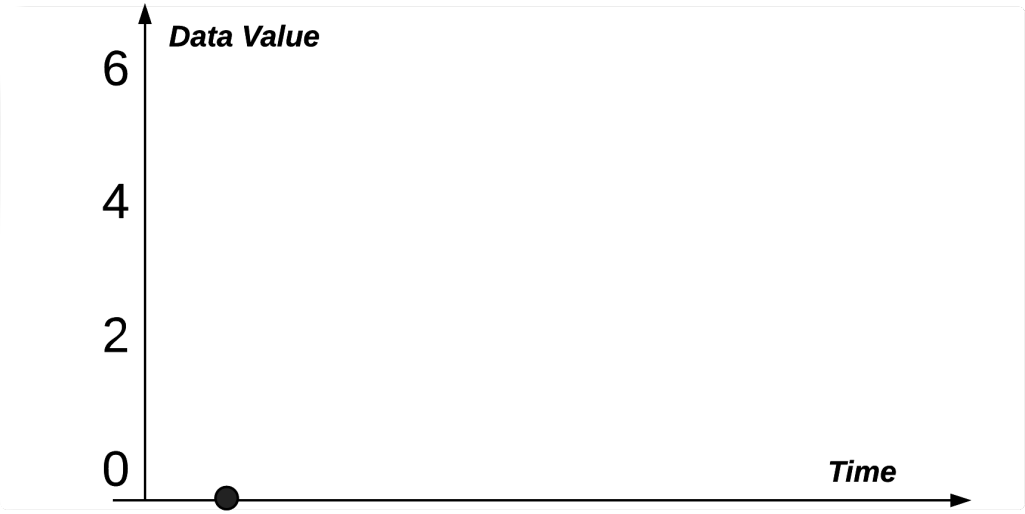


Tracking Component

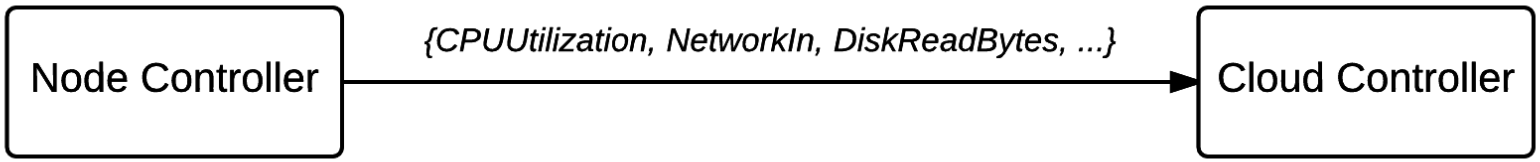


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

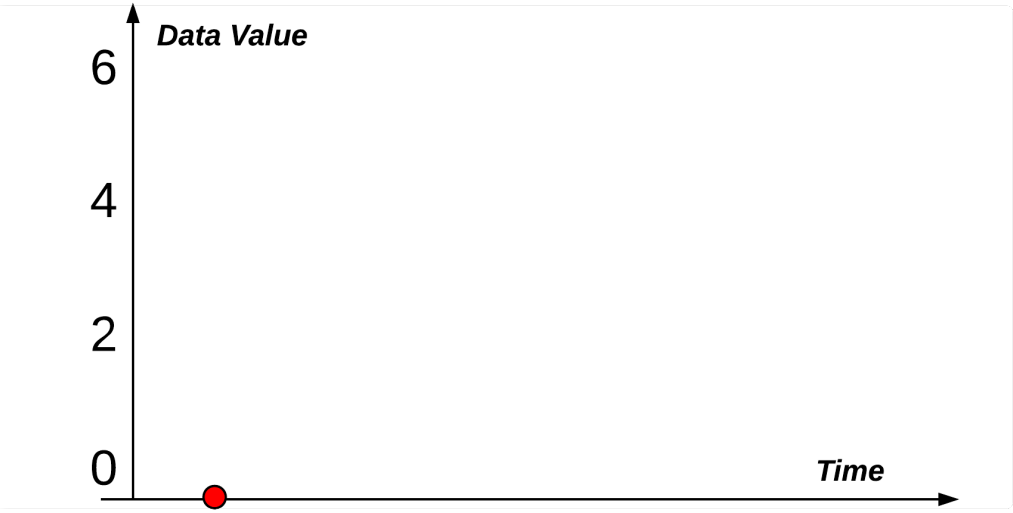


Tracking Component

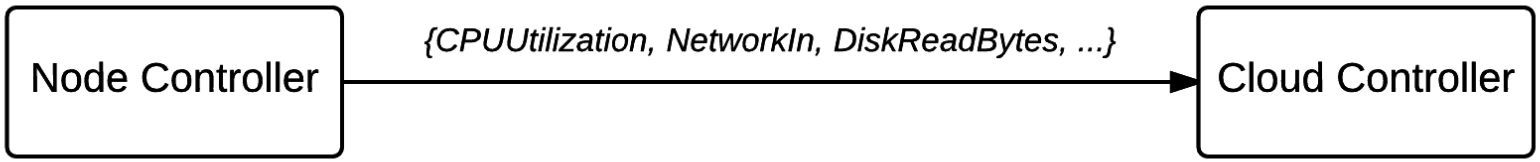


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

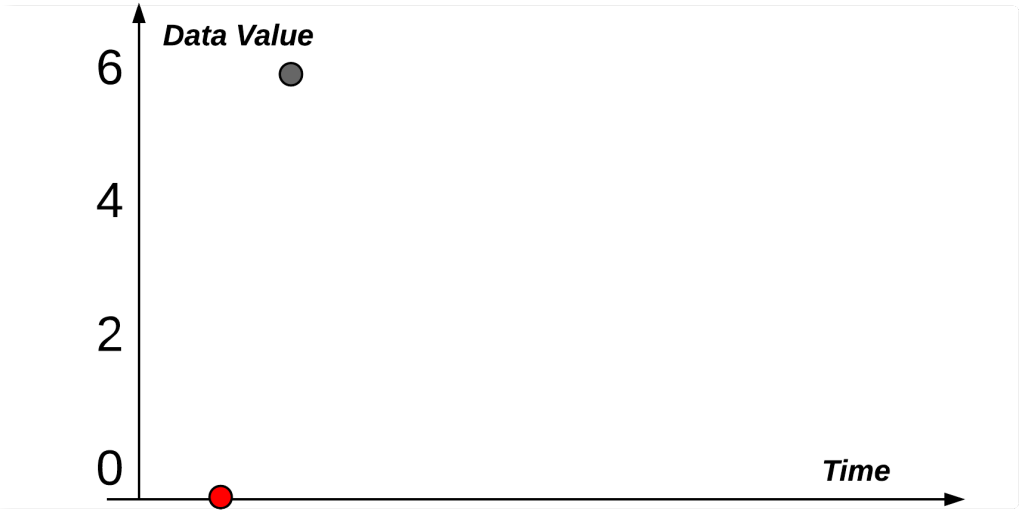


Tracking Component

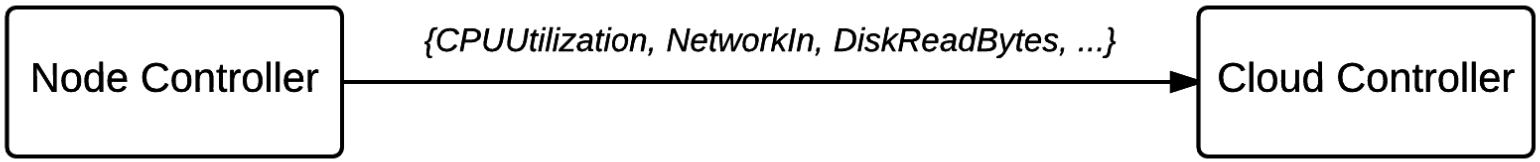


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

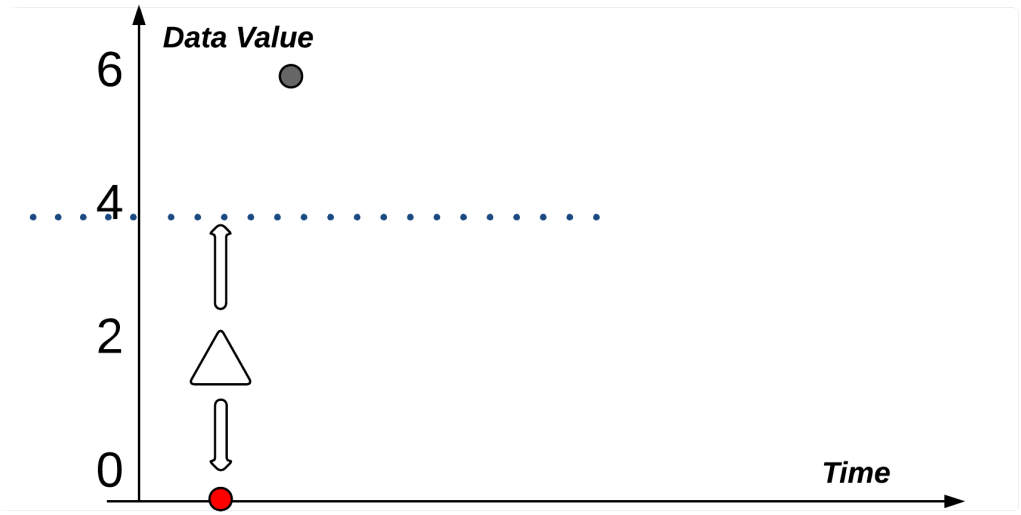


Tracking Component

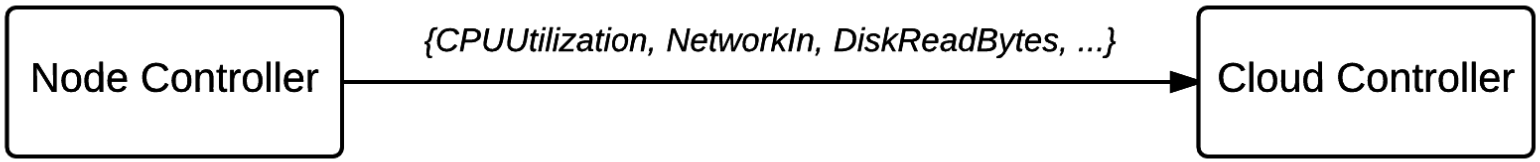


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

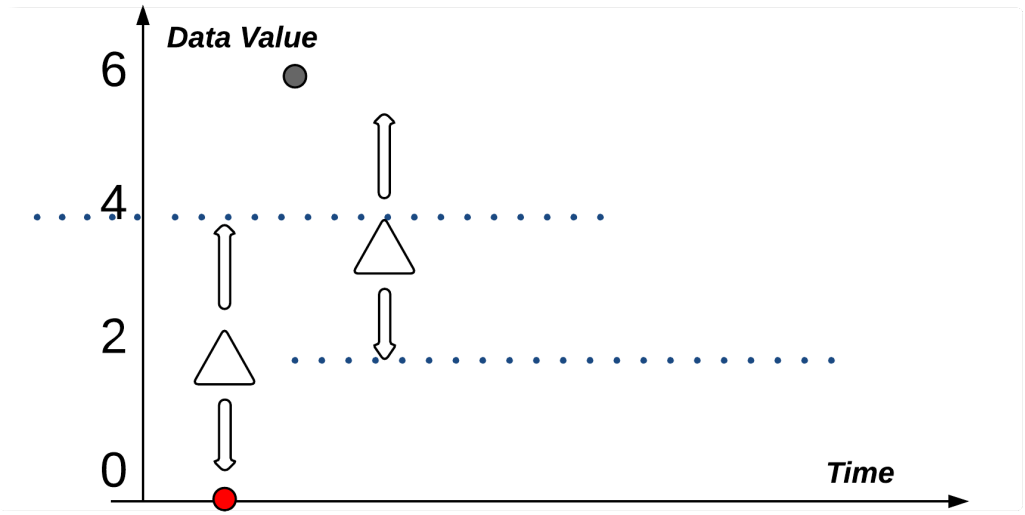


Tracking Component

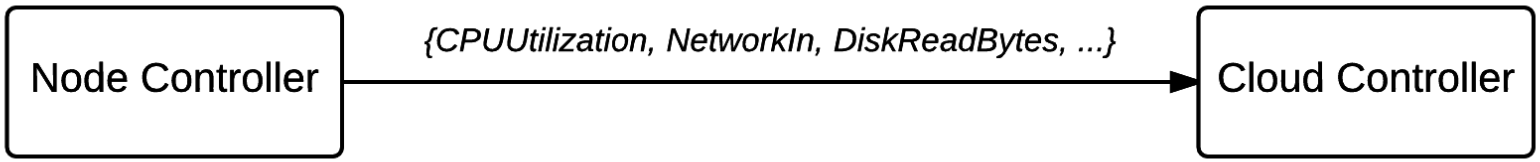


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

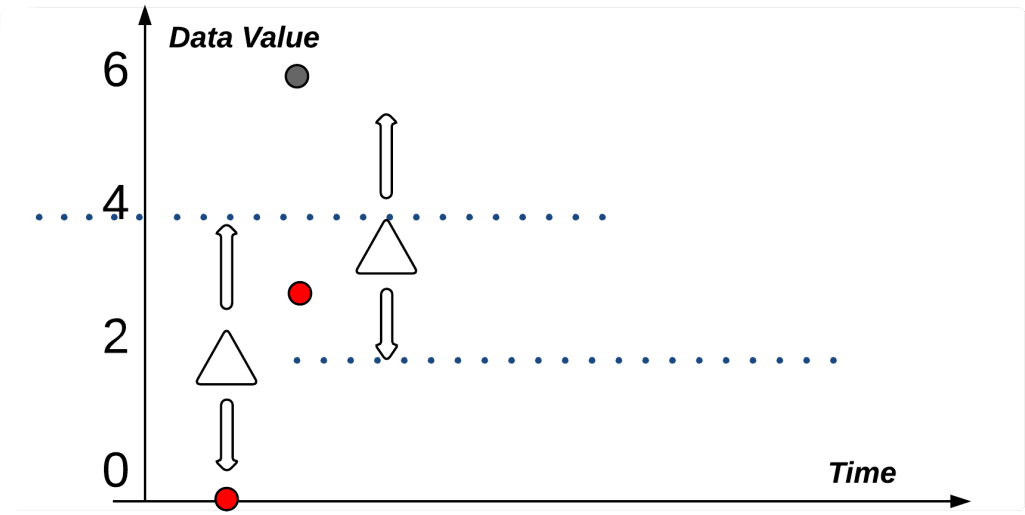


Tracking Component

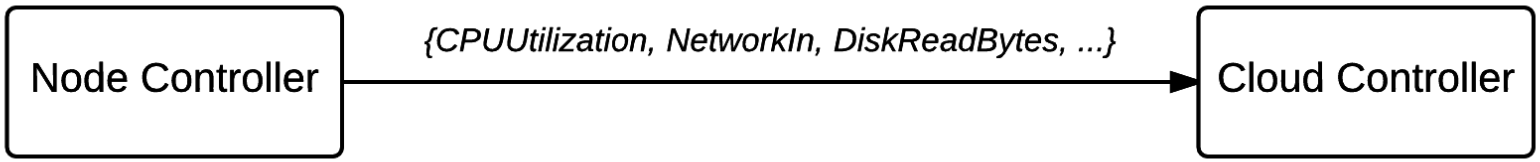


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

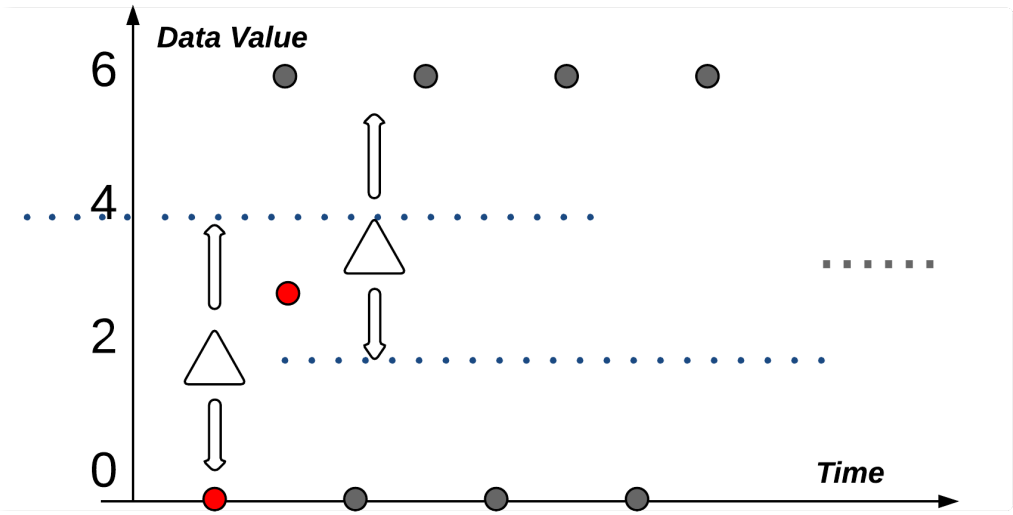


Tracking Component

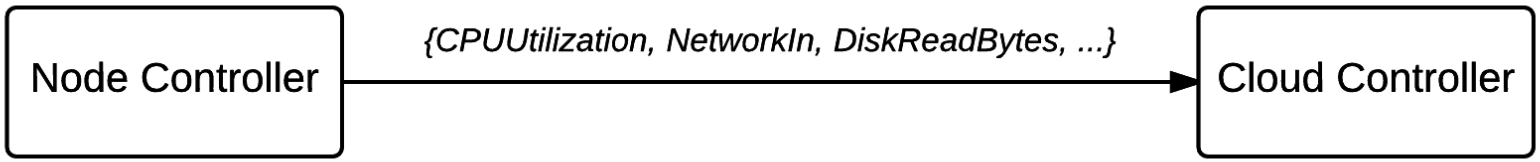


What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

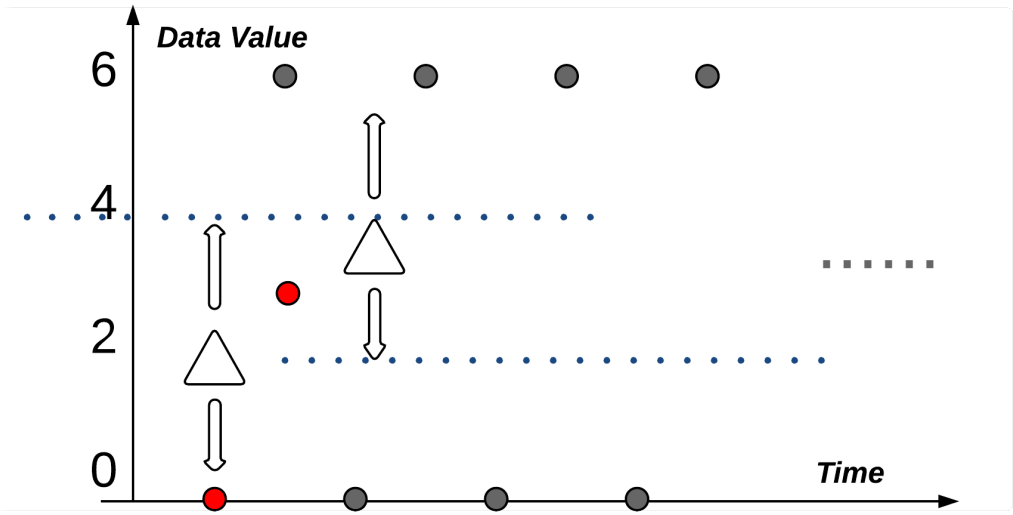


Tracking Component



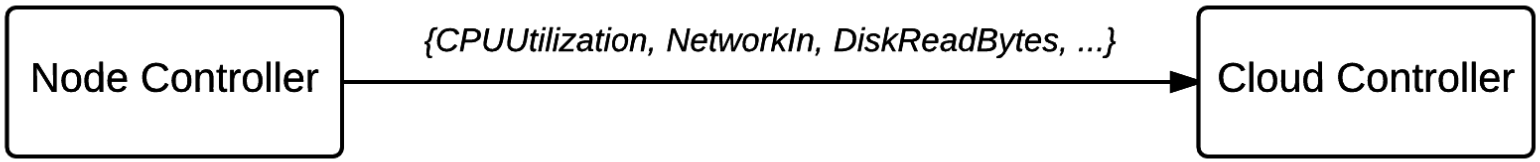
What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:



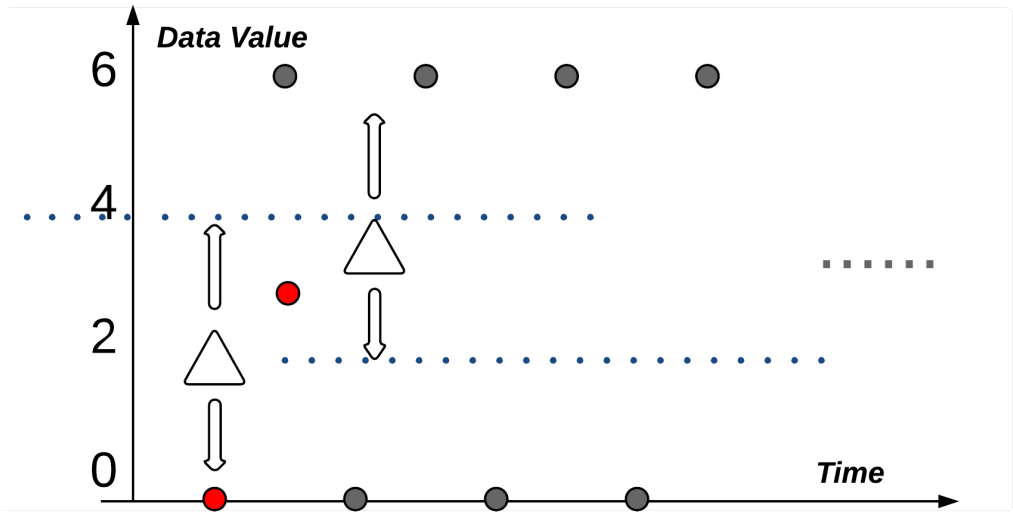
* Values sent: $\{0, 3\}$

Tracking Component



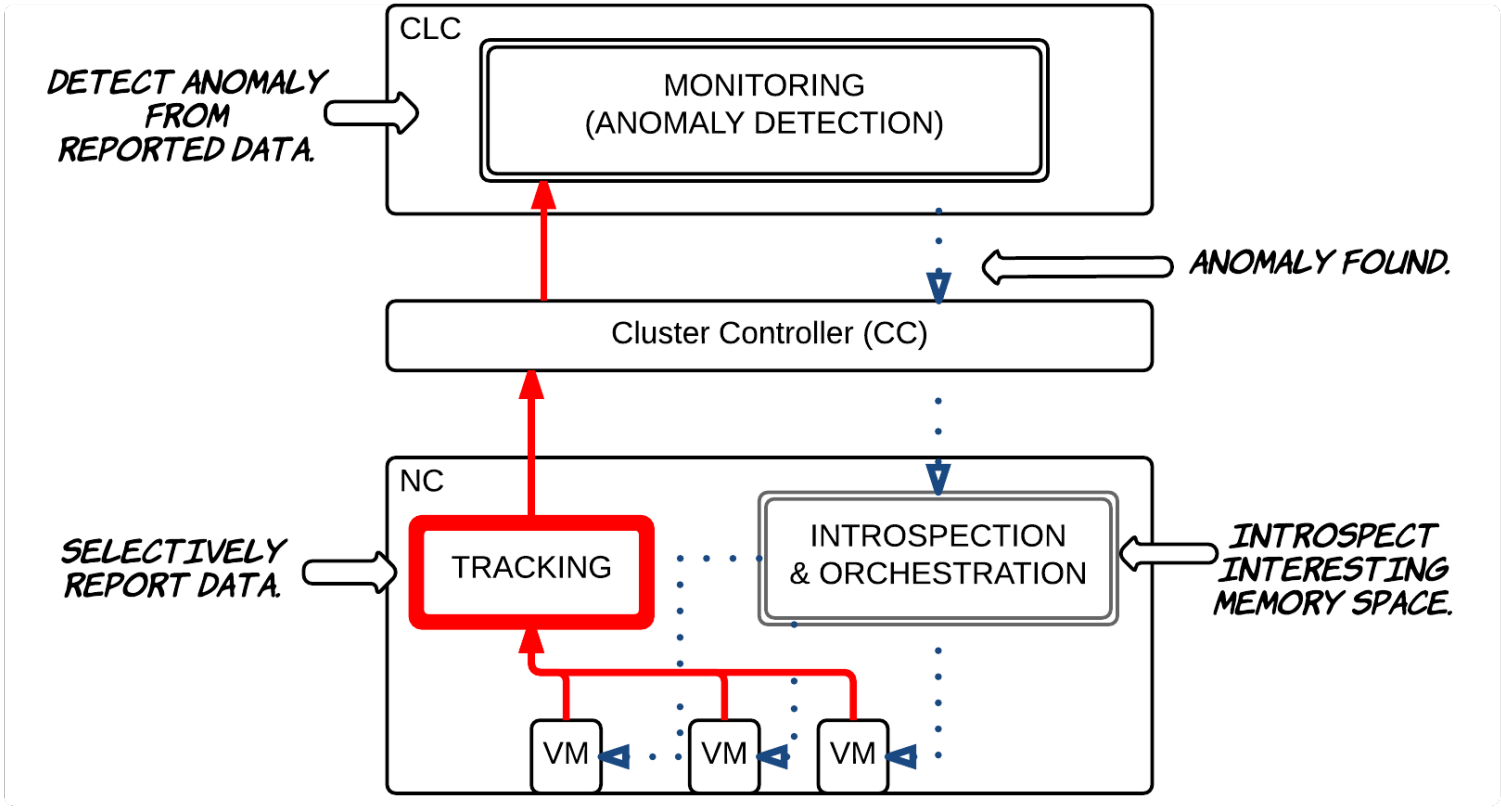
What if a small error Δ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

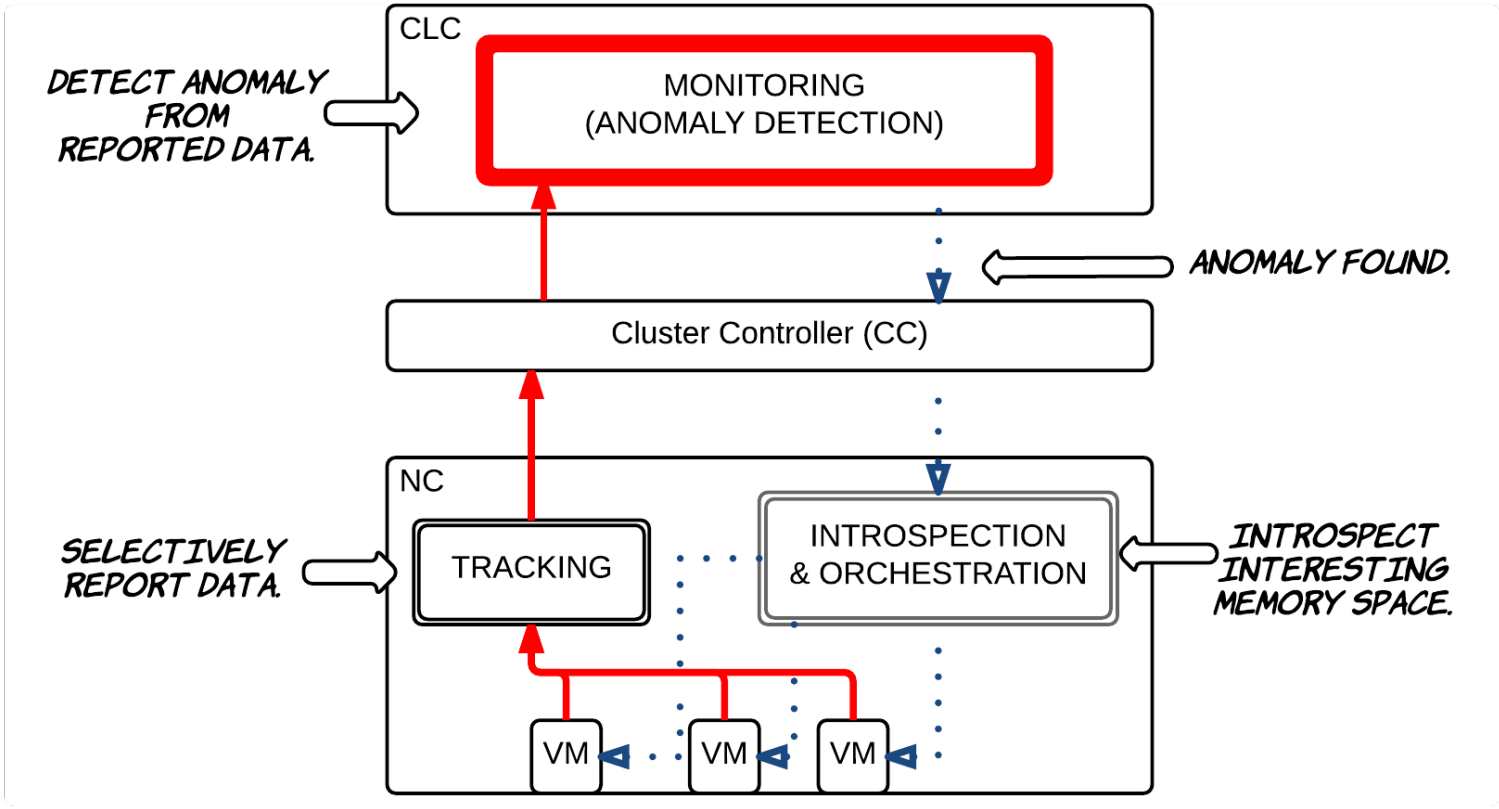


* Values sent: $\{0, 3\}$
* Competitive ratio: $\log \Delta$

Tracking Component



Monitoring Component



Monitoring Component

Monitoring Component

Data matrix reported from each node:

$$\underbrace{\left(\begin{array}{ccccc} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d} \end{array}\right)}_{d \text{ metrics}} \Bigg\} n \text{ time instances}$$

Monitoring Component

Data matrix reported from each node:

$$\underbrace{\left(\begin{array}{ccccc} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d} \end{array}\right)}_{d \text{ metrics}} \Bigg\} n \text{ time instances}$$

- Anomaly detection using this matrx;

Monitoring Component

Data matrix reported from each node:

$$\underbrace{\left.\begin{pmatrix} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d} \end{pmatrix}\right\} n \text{ time instances}}_{d \text{ metrics}}$$

- Anomaly detection using this matrx;
- Use Principal Component Analysis (PCA);

Monitoring Component

Data matrix reported from each node:

$$\underbrace{\left(\begin{array}{ccccc} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d} \end{array}\right)}_{d \text{ metrics}} \Bigg\} n \text{ time instances}$$

- Anomaly detection using this matrx;
- Use Principal Component Analysis (PCA);
- Sliding window;

Monitoring Component

Data matrix reported from each node:

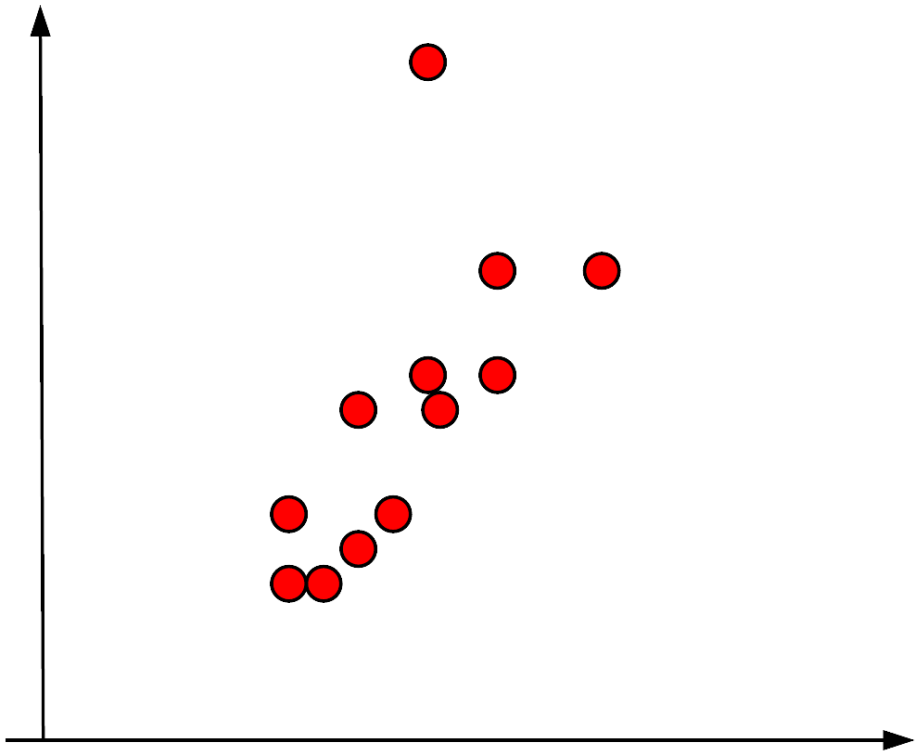
$$\underbrace{\left(\begin{array}{ccccc} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d} \end{array}\right)}_{d \text{ metrics}} \Bigg\} n \text{ time instances}$$

- Anomaly detection using this matrx;
- Use Principal Component Analysis (PCA);
- Sliding window;
- Metrics identification after anomalies are detected.

Monitoring Component - Anomaly Detection

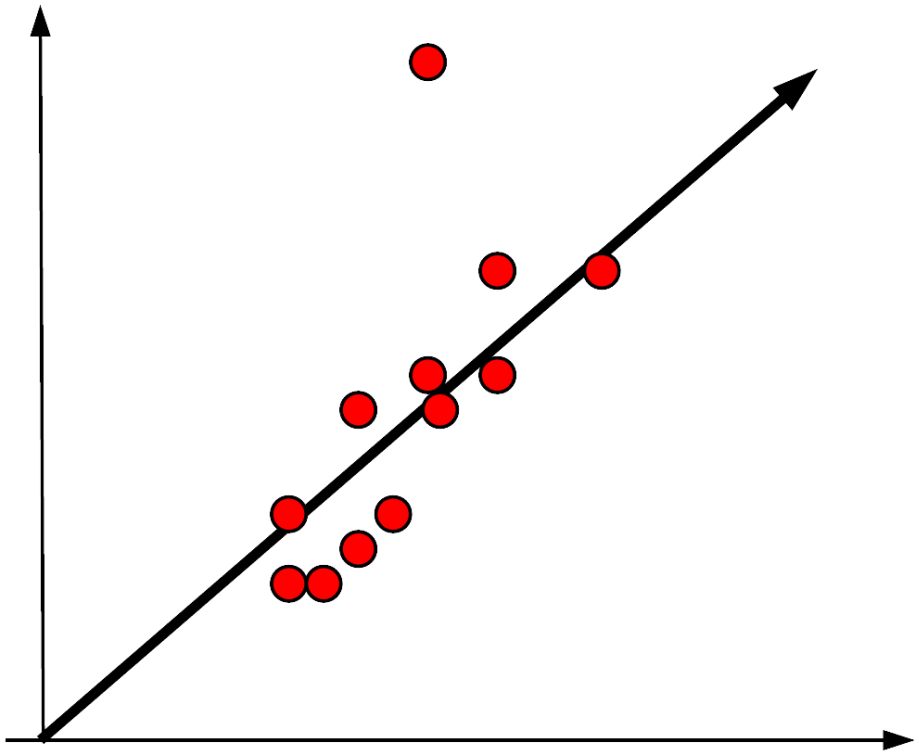
Monitoring Component - Anomaly Detection

PCA:



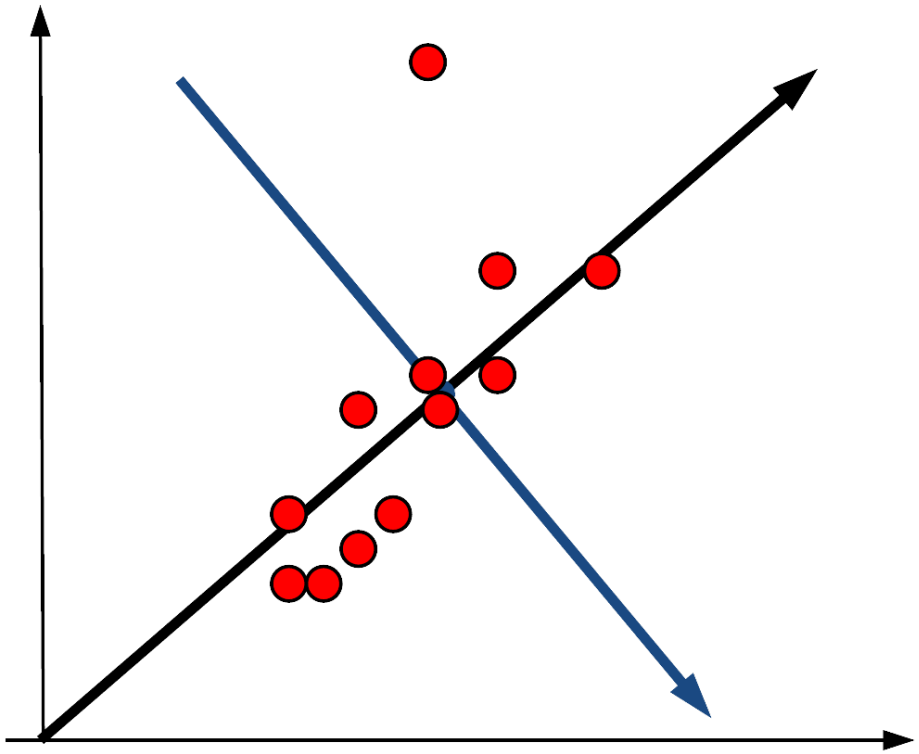
Monitoring Component - Anomaly Detection

PCA:



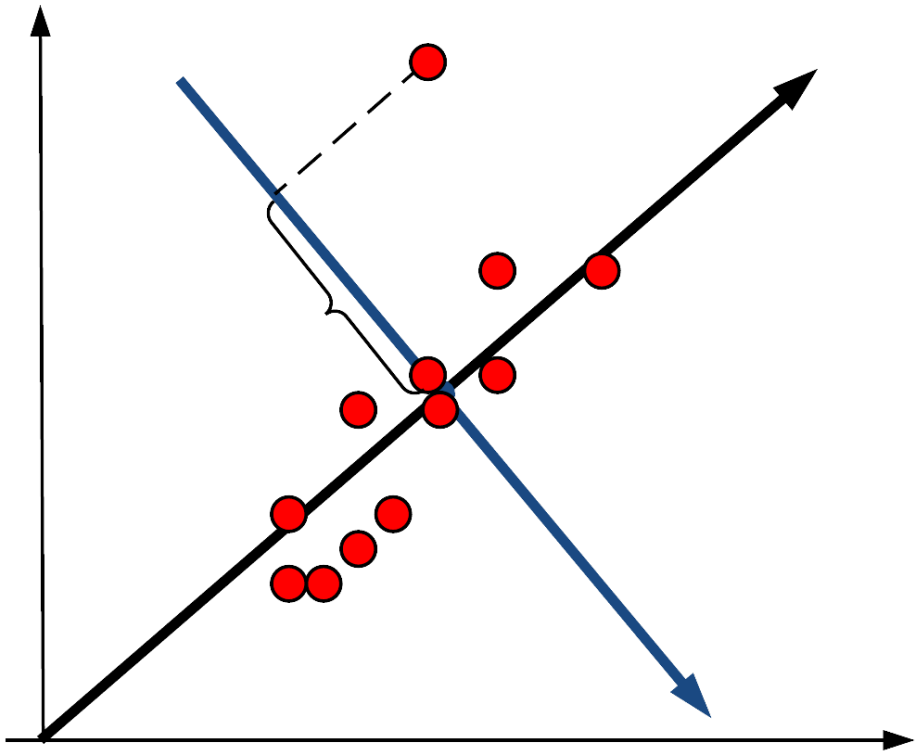
Monitoring Component - Anomaly Detection

PCA:



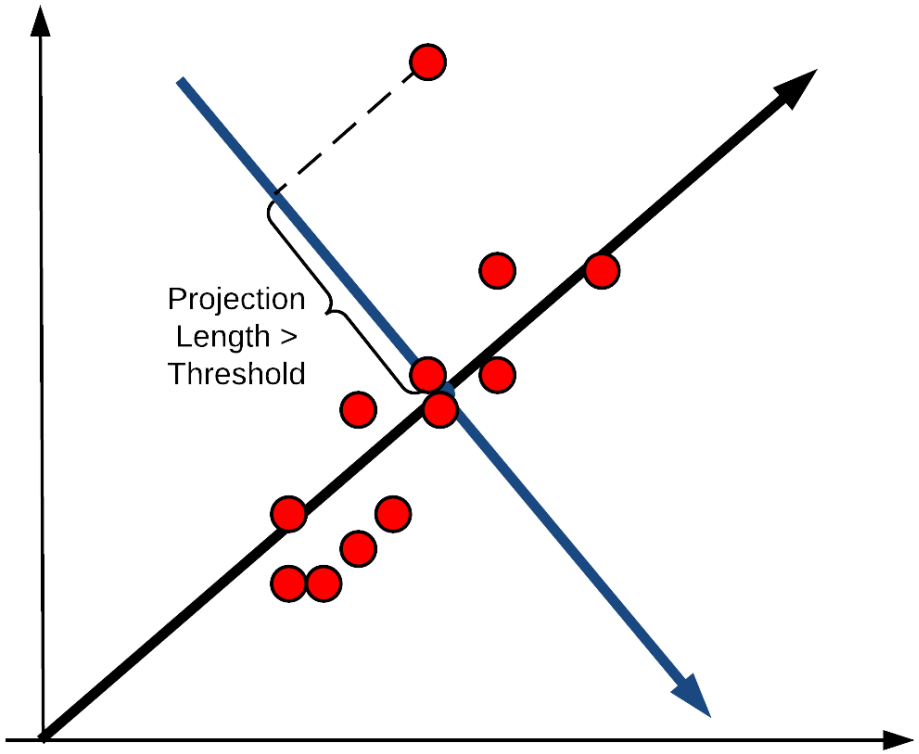
Monitoring Component - Anomaly Detection

PCA:



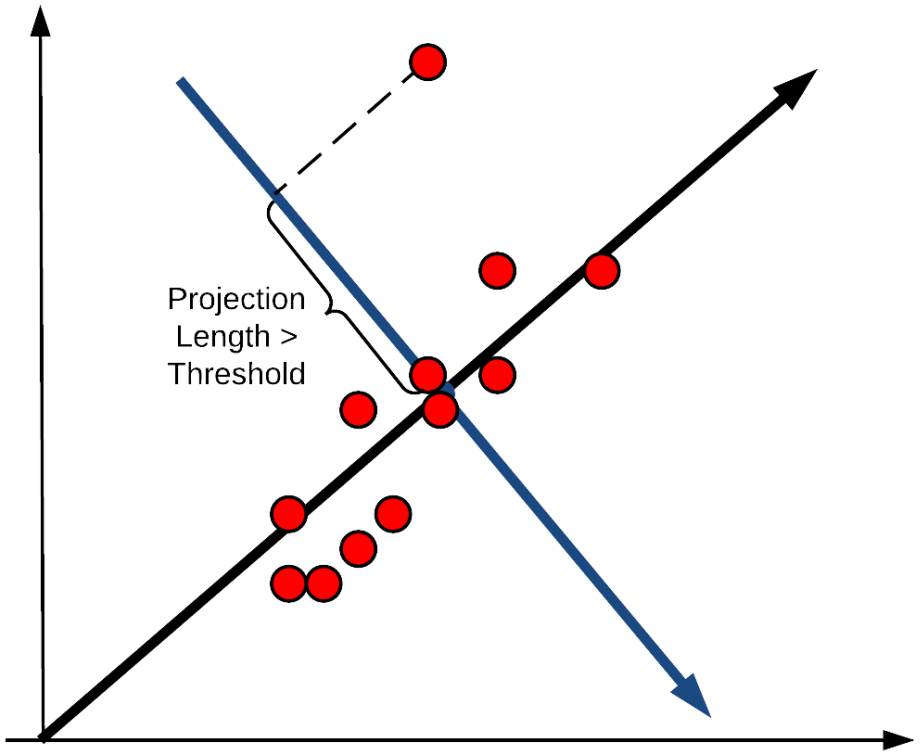
Monitoring Component - Anomaly Detection

PCA:



Monitoring Component - Anomaly Detection

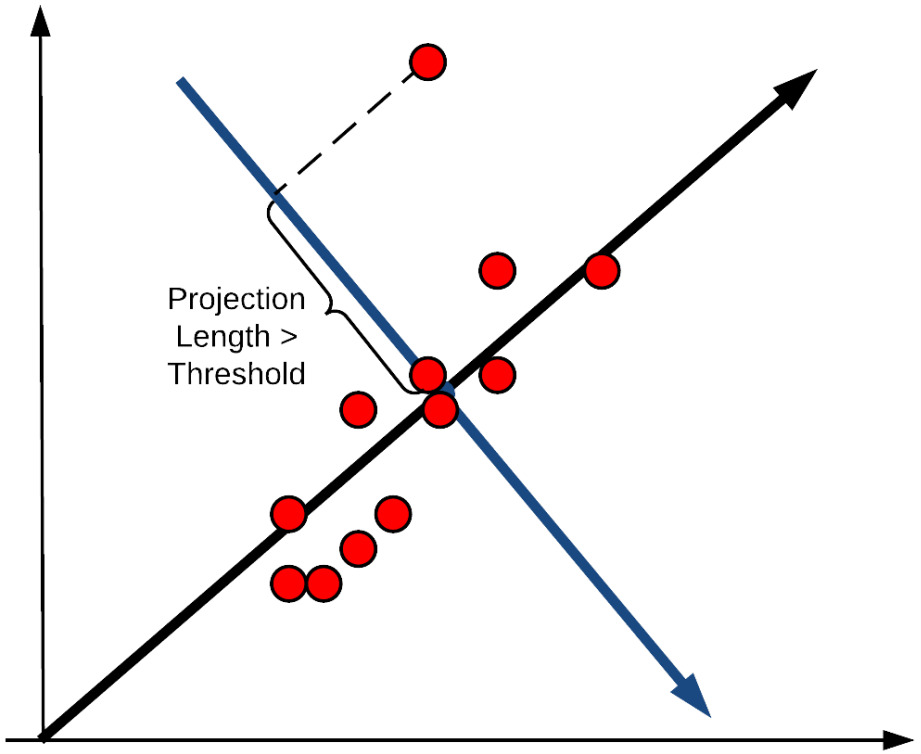
PCA:



+ Threshold Q_α is computed according to a given false alarm rate α.

Monitoring Component - Anomaly Detection

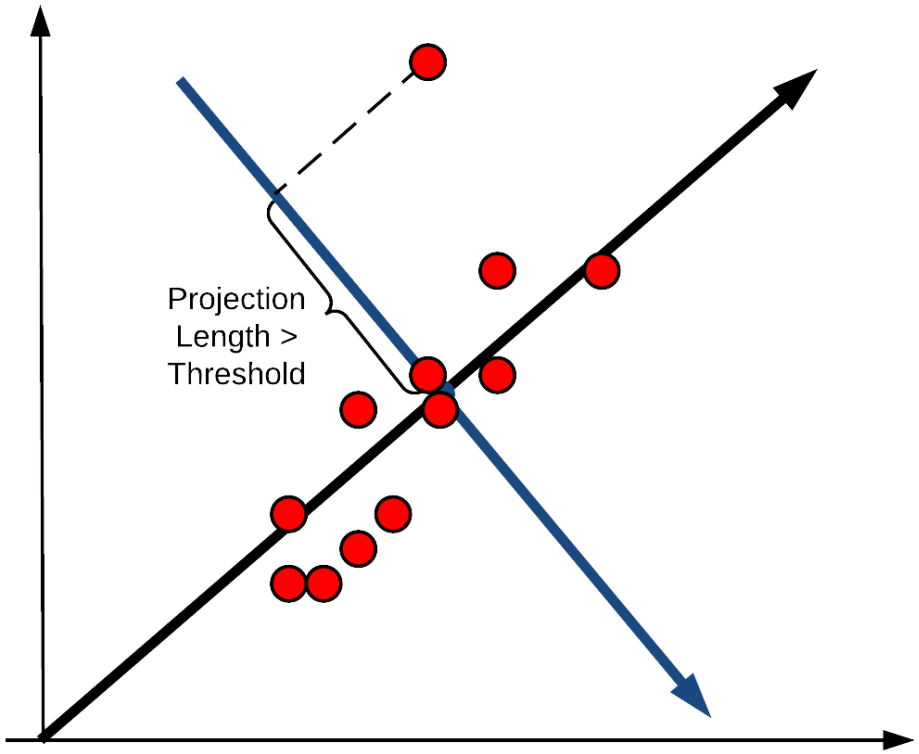
PCA:



- Threshold Q_α is computed according to a given false alarm rate α.
- Tracking component introduces error Δ to data matrix.

Monitoring Component - Anomaly Detection

PCA:



- Threshold Q_α is computed according to a given false alarm rate α.
- Tracking component introduces error Δ to data matrix.
- Given μ, dynamically adjust Δ according to PCA results, to ensure false alarm rate $\in (\alpha - \mu, \alpha + \mu)$

Monitoring Component - Metrics Identification

Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

$$\begin{pmatrix} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d} \end{pmatrix}$$

Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

$$\begin{pmatrix} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d} \end{pmatrix}$$

Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

$$\begin{pmatrix} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & \mathbf{V_{(now)1}} & \mathbf{V_{(now)2}} & \cdots & V_{(now)d} \end{pmatrix}$$

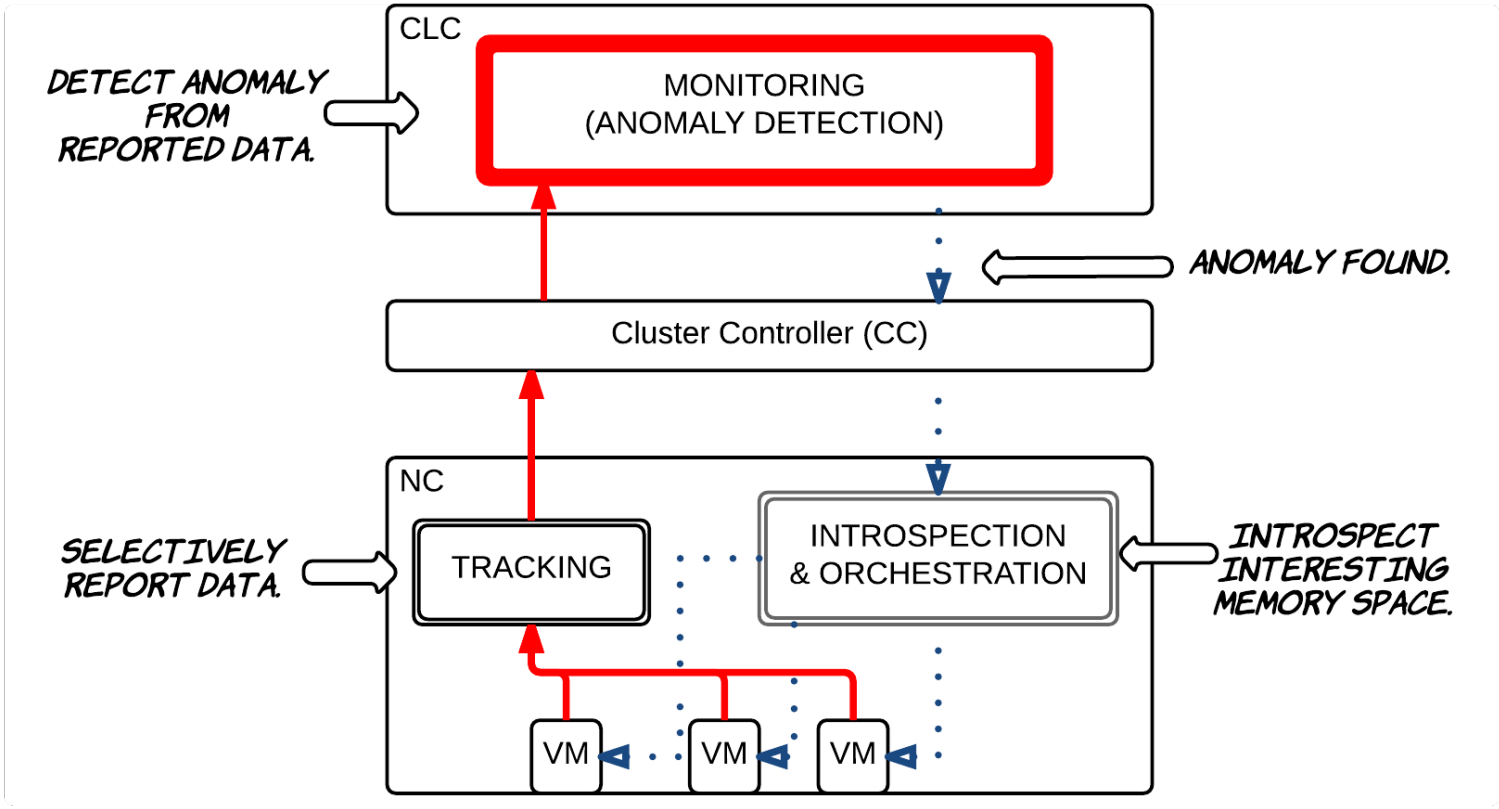
Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

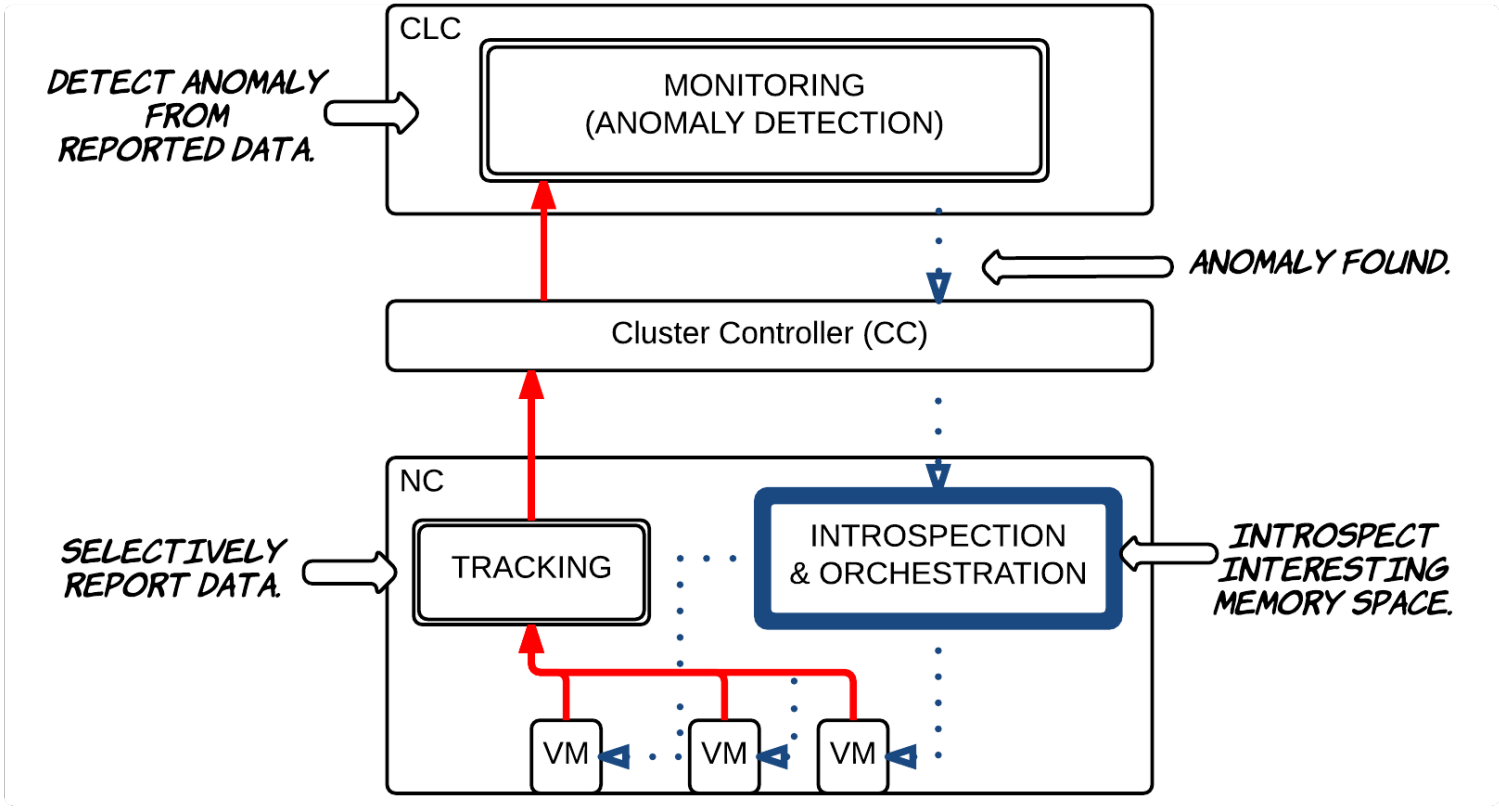
$$\begin{pmatrix} V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\ \vdots & & & \ddots & \\ V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\ V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\ V_{(now)0} & \mathbf{V_{(now)1}} & \mathbf{V_{(now)2}} & \cdots & V_{(now)d} \end{pmatrix}$$

Main idea: Compare each dimension of the abnormal data points and normal ones.

Monitoring Component



Orchestration Component



Orchestration Component

Orchestration Component

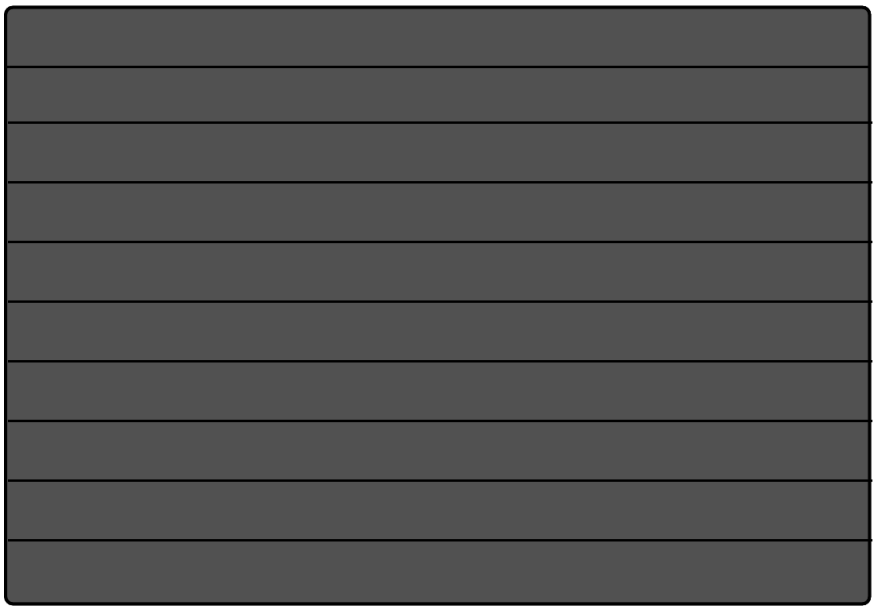
Virtual Machine Introspection (VMI)

- Introspect VM memory using existing VMI tools;

Orchestration Component

Virtual Machine Introspection (VMI)

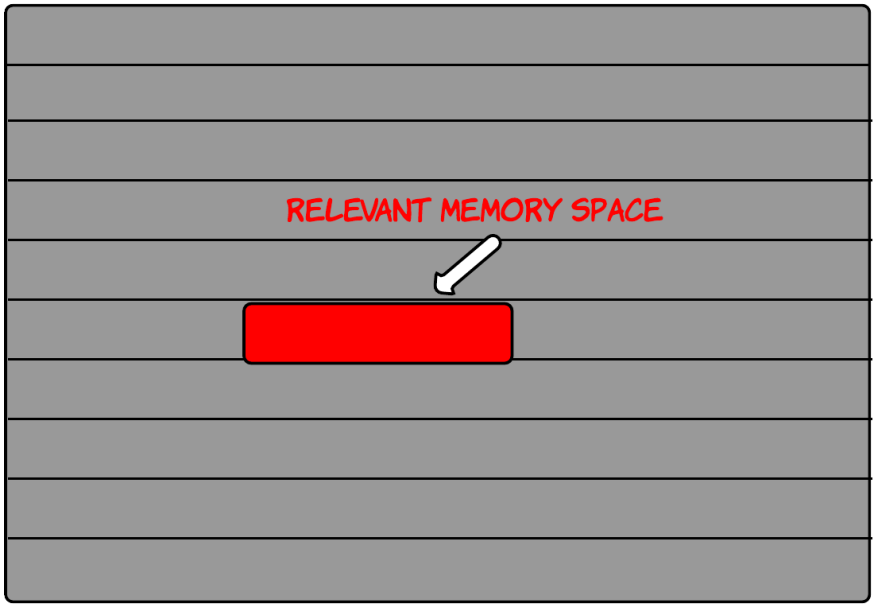
- Introspect VM memory using existing VMI tools;



Orchestration Component

Virtual Machine Introspection (VMI)

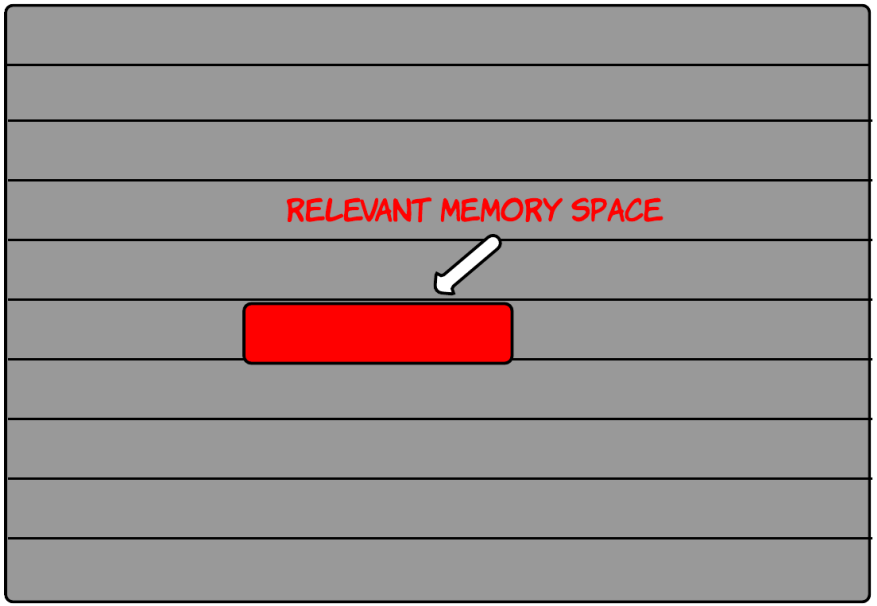
- Introspect VM memory using existing VMI tools;



Orchestration Component

Virtual Machine Introspection (VMI)

- Introspect VM memory using existing VMI tools;

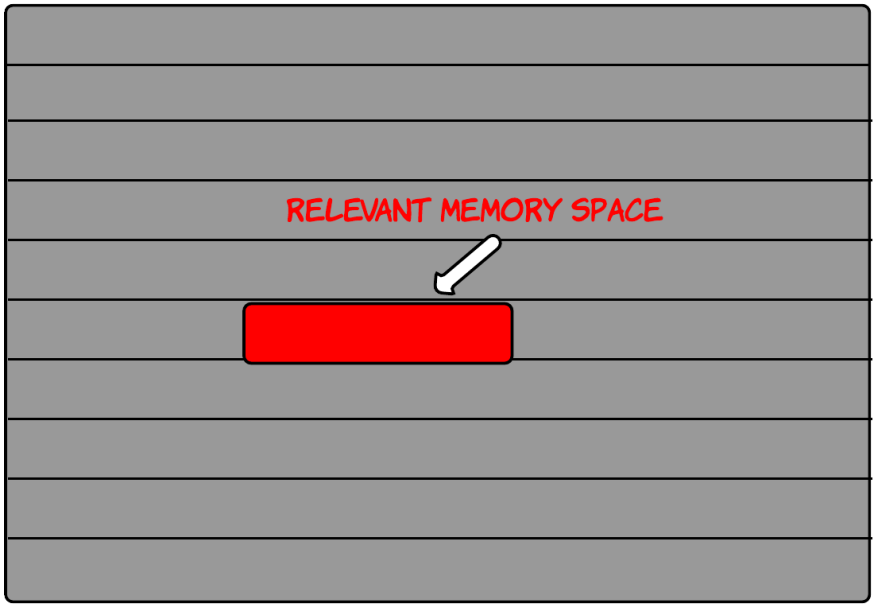


- Raise alarm;

Orchestration Component

Virtual Machine Introspection (VMI)

- Introspect VM memory using existing VMI tools;



- Raise alarm;
- Optionally, kill process.

Evaluation

+ Implemented on the Eucalyptus Cloud platform;

Evaluation

+ Implemented on the Eucalyptus Cloud platform;
+ Modified Node Controller and Cloud Controller source code.

Evaluation

Recall the two questions:

Evaluation

Recall the two questions:

1. Monitor more efficiently?

Evaluation

Recall the two questions:

1. Monitor more efficiently?

2. Utilize the statistics for security purpose?

Evaluation

Recall the two questions:

1. Monitor more efficiently?
   - Tracking Component
2. Utilize the statistics for security purpose?

Evaluation

Recall the two questions:

1. Monitor more efficiently?
   - Tracking Component
2. Utilize the statistics for security purpose?
   - Monitoring and Orchestration Component

Evaluation

Recall the two questions:

1. Monitor more efficiently?
   - Tracking Component
2. Utilize the statistics for security purpose?
   - Monitoring and Orchestration Component

Metrics monitored for each VM:

- The default 7 metrics monitored by Eucalyptus CloudWatch.

Evaluation - Tracking

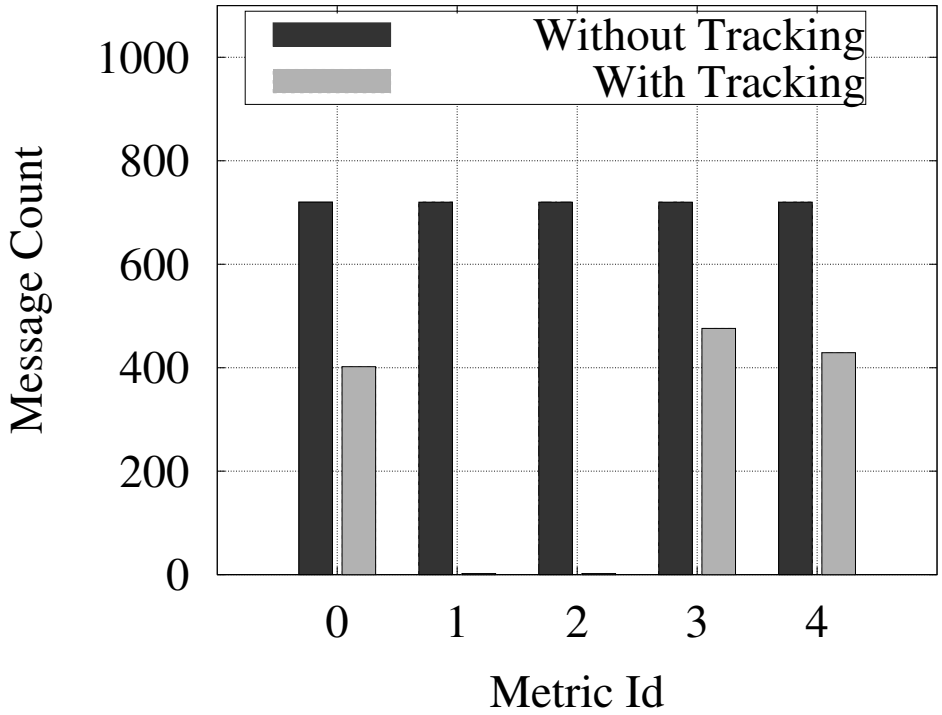
A comparison on number of values sent by NC for each metric.

- VM workload: TPC-C benchmark on MySQL database;
- Δ: The average for each metric when VM is **idle**.

Evaluation - Tracking

A comparison on number of values sent by NC for each metric.

- VM workload: TPC-C benchmark on MySQL database;
- Δ: The average for each metric when VM is **idle**.



Evaluation - Monitoring

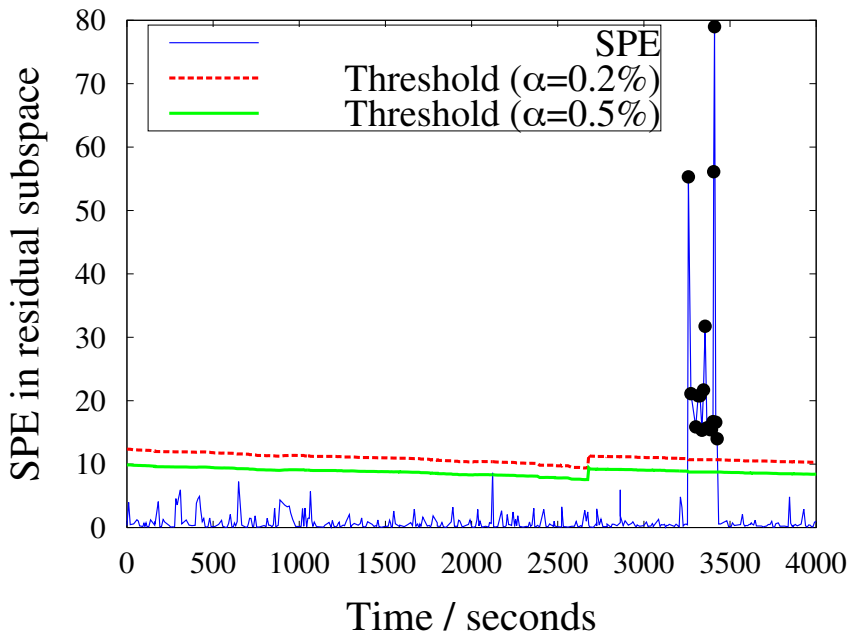
Experiment setting:

- 3 VMs being monitored: VM 1 idle, VM 2 and 3 run Apache web server;
- VM 2 and 3 are compromised as DDoS bots later.

Evaluation - Monitoring

Experiment setting:

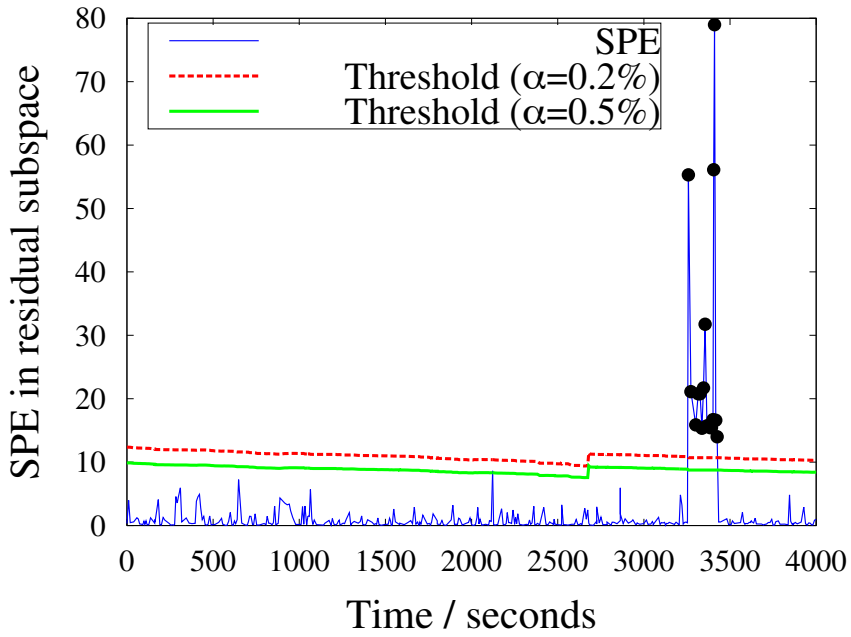
- 3 VMs being monitored: VM 1 idle, VM 2 and 3 run Apache web server;
- VM 2 and 3 are compromised as DDoS bots later.



Evaluation - Monitoring

Experiment setting:

- 3 VMs being monitored: VM 1 idle, VM 2 and 3 run Apache web server;
- VM 2 and 3 are compromised as DDoS bots later.



Dim (j)	vm1-d1	vm1-d2	vm1-d3	vm1-d4	vm1-d5	vm1-d6	vm1-d7	vm2-d1	vm2-d2	vm2-d3	vm2-d4
rd_j	23.70	-0.98	-0.98	-0.55	-0.57	4.27	3.76	9.14	64.18	65.05	3.50
$stddev_j$	0.78	*0.42*	*0.58*	*0.00*	*0.67*	0.00	0.71	**3.17**	**8.01**	**8.30**	0.00
$meandiff_j$								0.16	-0.26	-0.28	
Dim (j)	vm2-d5	vm2-d6	vm2-d7	vm3-d1	vm3-d2	vm3-d3	vm3-d4	vm3-d5	vm3-d6	vm3-d7	
rd_j	-0.51	-0.82	4.23	9.04	60.56	61.16	1.45	-0.56	1.89	-0.51	
$stddev_j$	*0.31*	*0.00*	0.35	**7.23**	**6.06**	**6.98**	*0.17*	*3.39*	*0.12*	*3.65*	
$meandiff_j$				0.39	-0.23	-0.31					

Metrics Identification Result

THE UNIVERSITY OF UTAH

Evaluation - Orchestration

- Received a VMI request with information:
  - A possible network problem;
  - Similar patterns for VM 2 and 3.

Evaluation - Orchestration

- Received a VMI request with information:
  - A possible network problem;
  - Similar patterns for VM 2 and 3.
- Node Controller call existing VMI tools to introspect:
  - VM 2: Volatility found suspicious DDoS process;
  - VM 3: Same with VM 2?
  - Raise alarm to user;
  - Kill the processes automatically using StackDB if confirmed.

Discussion - Overhead

Discussion - Overhead

Overhead introduced:

- On NC: $O(1)$ to apply tracking algorithm and call VMI tools;
- On CLC: Depending on the PCA algorithm used, polynomial to sliding window size and number of dimensions monitored.

Discussion - Overhead

Overhead introduced:

- On NC: $O(1)$ to apply tracking algorithm and call VMI tools;
- On CLC: Depending on the PCA algorithm used, polynomial to sliding window size and number of dimensions monitored.

Overhead saved:

- Significant amount of network traffic sending from NC to CC to CLC;
- Significant amount of memory space to be introspected by VMI.

Discussion - Possible Extension

Discussion - Possible Extension

- Monitor more metrics;
- Extend VMI tools to find more complicated attacks.

Thank you.

Thank you.

Questions?