Paracloud: Bringing Application Insight into Cloud Operations

Shripad J Nadgowda, Sahil Suneja, Canturk Isci

IBM T J Watson Research Center
Agenda

• (Historical) Background
• Motivation (for Change)
• Paracloud Architecture (What, Why and How?)
• (Validation by) Evaluation
• (Take away) Conclusion
• (Open for) Discussion
Applications were traditionally designed and developed to run on physical servers.
Tale of Application development

Application Development

Application Deployments

Applications on-boarded to IaaS Clouds in VMs and remained un-modified
Tale of Application development

Application Development

Application Deployments

Containers

IaaS Cloud

PaaS Cloud

Applications on-boarded to PaaS clouds encapsulated in container images
Tale of Application development

1. Created 1:N relationship between application developments and their deployment types
2. Applications have been oblivious to their actual runtimes
Lets flip the coin...
Tale of Application runtimes

A single generic operating runtime to host different kinds of applications
Tale of Application runtimes

A *single generic* hypervisor to host different *kinds* of application VMs
Tale of Application runtimes

Physical server runtime

Virtual server runtime

- JavaScript
- MySQL
- NGINX

Host OS

HW

CPU  Memory  IO

Apps

fadvise
madvise
directIO
...

VM

Host OS

PVD

Apps

Hypervisor

guesttools

virtIO

...

HW

CPU  Memory  IO
Tale of Application runtimes

Applications know best!
A single generic cloud platform to host different kinds of application containers
Tale of Application runtimes

A *single generic* cloud platform to host different *kinds* of application containers.
Tale of Application runtimes

PaCI: Paracloud Interface for *modular* Cloud operations
Credit: Land! Boat! I'm Saved! By Sean Kleefeld
Paracloud: What?

- Applications are run directly on top of cloud platforms.
- Paracloud is a framework in which a bi-directional control channel (PaCI) is available between applications and cloud substrate.
- PaCIs are modular extensions to the cloud management functions like migration, auto-scaling, load-balancing etc.
- Currently, PaCIs are designed and being implemented for Kubernetes.
Paracloud: Why?

**Migration**
- Ensure consistency across in-memory state and persistent state
- Reduce memory pressure during migration

**Load-Balancing**
- Ability to co-operatively load-balanced across different kinds of services
- Build a semantically equivalent to linux’s *yield* for cloud

**Auto-scaling**
- Eliminate false-positive scaling triggers
- Ability to select the right scaling methods
Paracloud: How?

Cloud Platform: Kubernetes
- Most popular container cloud substrate
- Supports container hooks and Downward APIs
- Vision for cluster-native applications

<table>
<thead>
<tr>
<th>PaCI</th>
<th>Delivery Guarantee</th>
</tr>
</thead>
<tbody>
<tr>
<td>preMigrate</td>
<td>At-most-once</td>
</tr>
<tr>
<td>postRestore</td>
<td>At-most-once</td>
</tr>
<tr>
<td>reqYield</td>
<td>At-least-once</td>
</tr>
<tr>
<td>appYield</td>
<td>–</td>
</tr>
<tr>
<td>chkAutoscale</td>
<td>At-least-once</td>
</tr>
<tr>
<td>hotScale</td>
<td>–</td>
</tr>
</tbody>
</table>

PaCI implemented as container hooks    implemented as extension to K8S APIs

Enforcement
Side-car containers w/o application change

Image Credits: dramaworkhouse.org.uk
Paracloud: Evaluation

HotScale: To be or not to be!

Applications-assisted auto-scaling improves performance by ~20%!
Conclusion

- Applications have been oblivious to their cloud runtimes
- Container cloud poses an opportunity to make applications cloud-aware
- Paracloud is a framework that enables such cloud-awareness for apps and app-awareness for clouds
- PaClIs currently defines bi-directional control interface for migration, auto-scaling and load-balancing
Discussion

Are these PaCIs secure?
• Currently, they are not security-profiled
• Adoptable for single-tenant cloud

How disruptive is it?
• It’s a controlled one😊

Whom does it matters the most?
• It’s a win-win for both application developer and cloud provider

Open Questions
• Should PaCI be vendor-agnostic, perhaps via a signal-and-syscall implementation?
• What are the minimum viable number of PaCIs to cover majority of cloud use cases?
• Whether PaCIs are applicable for non-containerized deployments
Q & A
Thank You

Contact: nadgowda@us.ibm.com