KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA


# How Lombard Odier Deployed VPA to Increase Resource Usage Efficiency

Vincent Sevel, Lombard Odier SA

# How Lombard Odier Deployed VPA to Increase Resource Usage Efficiency

KubeCon CloudNativeCon Europe 2022

Efficiency
/ɪˈfɪʃ(ə)nsi/
*noun*

the ratio of the useful work performed by a machine or in a process to the total energy expended or heat taken in.

"the boiler has an efficiency of 45 per cent“

Credit: Oxford Languages


Vincent Sevel
Architect / Platform Ops
v.sevel@lombardodier.com

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

# Lombard Odier Group

- Private Bank in Switzerland since 1796

- Main businesses
  - Private Clients
  - Asset Management
  - Technology for Banking

- Technology
  - Financial Software Solution Developer
  - BPO activity «Bank as a service»


LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

KubeCon CloudNativeCon Europe 2022

# Banking Platform





- 4 functional development streams
  - Market, Front, Tax & Operations, Finance
- Modular Service oriented solution
  - ≈ 800 application components
- GX: Large Modernization Initiative started in 2020
  - Functional &Technical

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

**0/27 nodes are available: 19 Insufficient cpu**

# Goal

Your Worker Node

- Optimize placement of pods in a Kubernetes cluster
- Tune resources on worker nodes
- Size optimally the underlying hardware
- Avoid waste
- Save money
  - … without sacrificing behavior

legacy static deployments vs K8s dynamic workloads

# Lombard Odier Clusters



| Cluster | Applicative Pods | Worker Nodes |
|---|---|---|
| Development | 600 | 19 |
| Test/UAT | 680 | 20 |
| Prod | 300 | 10 |

- Worker
  - 64 Gb
  - 20 cores
- ESX
  - 1 to 1.5 Tb of RAM
  - 72 to 96 physical cores
  - CPU overcommit ratio: 5

Eventually 20 to 30K pods
Can’t deal with resources by hand

# Resources on a container


- Request
  - Minimum required for placement
  - Maps to cpu-shares in docker
  - «schedule time»
- Limits
  - Maximum resource for a container
  - «run time»

- CPU
  - Compressible
  - Usage > limit || node = 100% => Throttling
- Memory
  - Not compressible
  - Usage > limit => Pod OOM
  - Node = 100% => Node OOM => Evictions (QoS)

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

.

# The right balance


- Oversizing requests
  - High behavior predictability
  - Low density
  - Low efficiency
  - May be compensated by virtualization overcommit
  - «Performance optimized»
  - Will steal extra capacity
- Undersizing requests
  - Low behavior predictability
    - CPU throttling
    - Pod eviction
  - High density
  - High efficiency
  - «Cost optimized»
  - May be starved from extra capacity

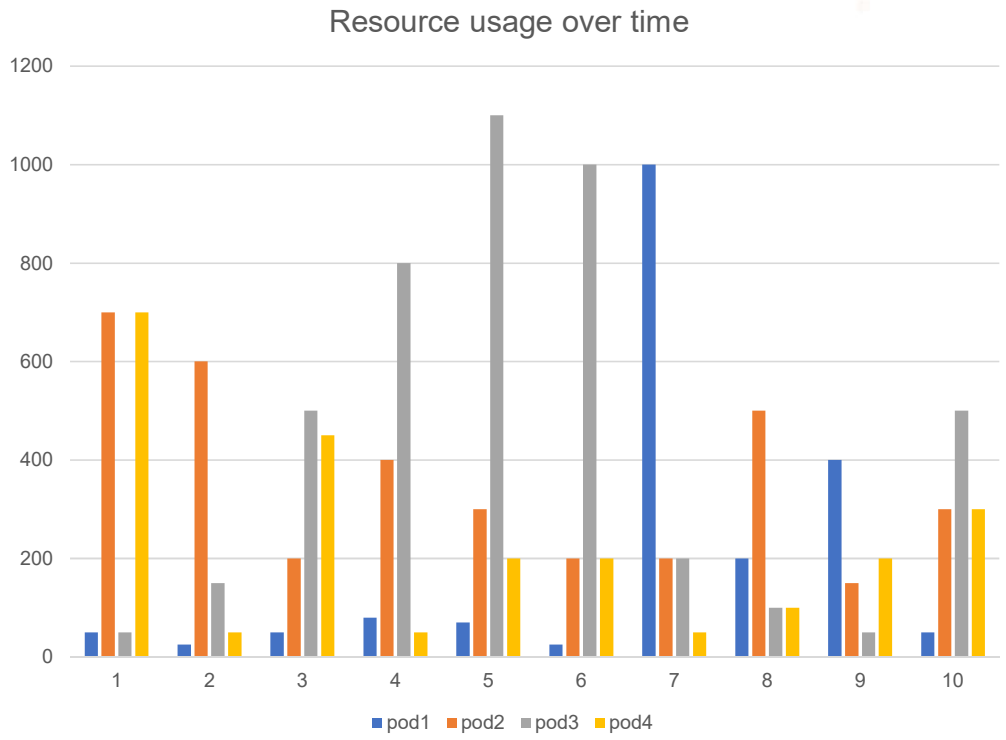
Assessing the requests?
Look at Past Behavior …

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

# Assessing the requests?

KubeCon CloudNativeCon Europe 2022

- Case study
  - 4 pods
  - 10 points in time
  - Resources from 25 to 1100
- What is the optimal worker node?
- What do pods need to reserve?
  - Avg? Max? Other?



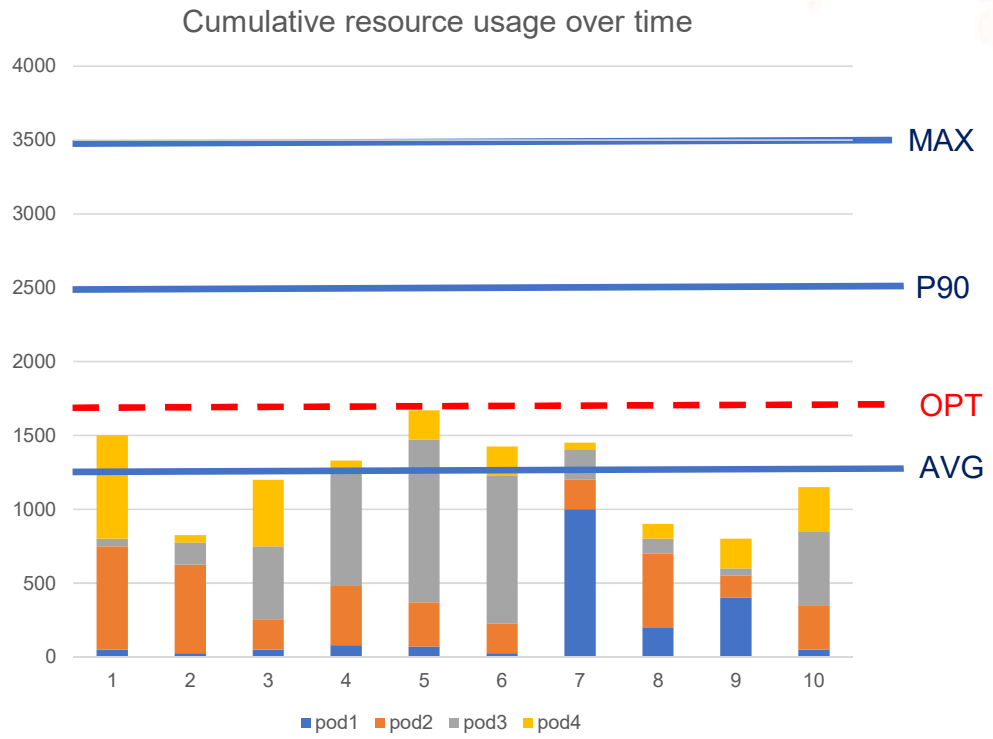

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

# Simulation results


- Optimal host = 1700
- If the requests were based on:
  - Sum(**Avg** Pods) = 1200
  - Sum(**P90** Pods) = 2500
  - Sum(**Max** Pods) = 3500

High density
Low predictability

Low density
High predictability




Pod1 Request_P90=460 Max=1000

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

# Vertical Pod Autoscaler (VPA)


- K8s subproject
- Recommend up-to-date resource limits and requests
- CRD based
- Watch & Store metrics
- Down-scale or Up-scale
- Memory & CPU
- Apply the recommendation (Opt)
- React to OOM

- Available in public clouds and on-premise
  - E.g. GKE autopilot, EKS, OpenShift Operator

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

# Example

```yaml
apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
…
spec:
  resourcePolicy:
    containerPolicies:
      - containerName: container-advisory-facade-jvm
        controlledResources:
          - cpu
          - memory
        controlledValues: RequestsOnly
        mode: Auto
  targetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: advisory-facade-jvm
  updatePolicy:
    updateMode: Initial
```

RequestsOnly
RequestsAndLimits

Off
Initial
Recreate
Auto

```yaml
status:
  conditions:
    - lastTransitionTime: '2022-05-05T09:25:34Z'
      status: 'True'
      type: RecommendationProvided
  recommendation:
    containerRecommendations:
      - containerName: container-advisory-facade-jvm
        lowerBound:
          cpu: 25m
          memory: '323371193'
        target:
          cpu: 25m
          memory: '351198544'
        uncappedTarget:
          cpu: 25m
          memory: '351198544'
        upperBound:
          cpu: 28m
          memory: '433310234'
```

```yaml
kind: Deployment
requests:
  cpu: 50m
  memory: 384Mi
```

```yaml
kind: Pod
requests:
  cpu: 25m
  memory: '351198544'
```

335Mb

.

# Use Cases


- **Stateless workloads**
- **(Cron)Jobs**
- Stateful workloads

- Limitations
  - JVM-based workloads (memory)
  - VPA vs HPA
  - Auto/Recreate with 1 pod
  - Excessive recommendations
  - Only 1 VPA object per workload
  - Number of VPA objects

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

# Where are we?



- Deployed on all clusters
- RequestsOnly

- CPU: **Initial**
  - Default request = 100m
  - Default Limit = 2 cores
  - Dev avg requ. 164m vs recomm. 42m
  - Dev: 70 cores saved (600 pods)
  - Test: 48 cores saved (680 pods)

- Memory: **Off** (except Dev: Initial)
  - Limit set by product team

- Capacity planning governance
  - Track usage vs requested
  - Target 50%  CPU density on Workers

| | Memory Request | Memory Usage | CPU Request | CPU Usage |
|---|---|---|---|---|
| Dev | 66% | 43% | 28% | 8% |
| Test | 80% | 60% | 25% | 6% |
| Prod | 73% | 61% | 36% | 5% |

# Lessons Learned


- NS/Object deletion + recreation
- 1 VPA object per Deployment
- Surprised by the low CPU recomm.
- ESX overcommit vs Pod request
- JVM workload, beware of
  - w/o VPA: Oversizing to cope with startup
  - Thundering herd problem: Startup on small clusters

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

# Next?


- Assess Memory/Initial
- Densify
- … Bare Metal
- Mix VPA with HPA/serverless
- Expand VPA to Third Party packages

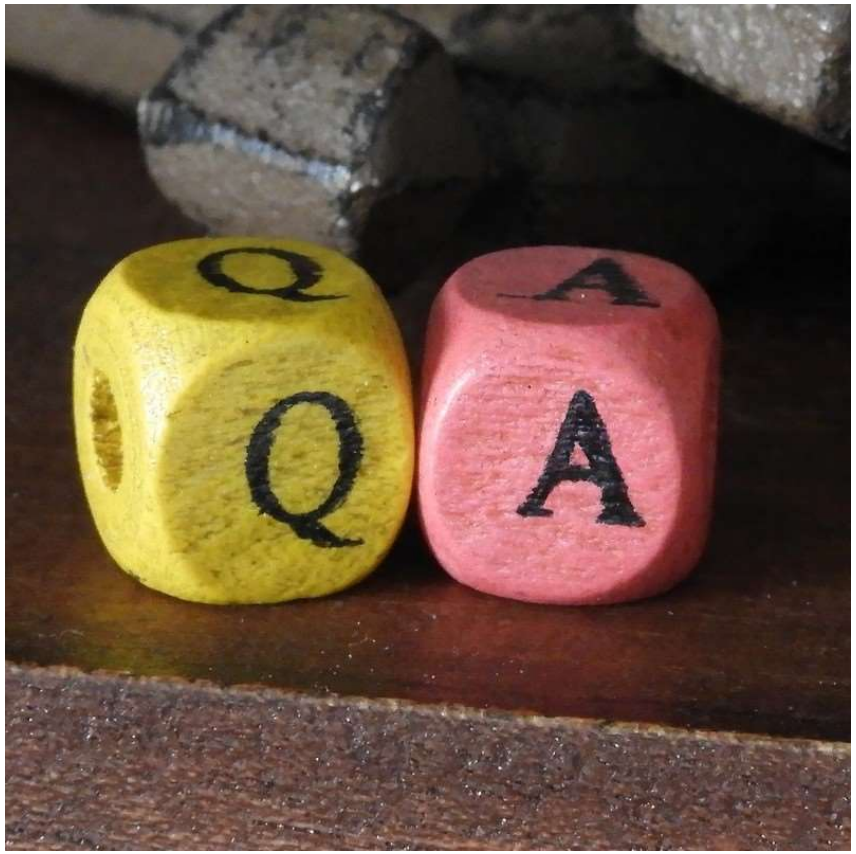
- move targetCPUPercentile into a flag (K8s/autoscaler #4799) by @matthyx
- In-Place Update of Pod Resources (K8s #1287) by @vinaykul
- Containers startup throttling (K8s #3312) – Closed
- Only one container policy used for recommendation (k8s/autoscaler #4861)

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH

# Tools and Resources


- kubectl vpa-recommendation plugin
- Goldilocks: An Open Source Tool for Recommending Resource Requests
- Harness: Optimize Kubernetes Costs with Resource Recommendations
  - Cost vs performance
- Must read: https://povilasv.me/vertical-pod-autoscaling-the-definitive-guide/

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH




KubeCon CloudNativeCon Europe 2022

A big thanks to Lombard Odier and the Platform Ops Team

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH



LOMBARD ODIER

LOMBARD ODIER DARIER HENTSCH

LOMBARD ODIER
LOMBARD ODIER DARIER HENTSCH