Serverless endpoint with public container clusters for HEP experiments
Belle II experiment run on SuperKEKB particle accelerator in Japan
Belle II cloud solutions for the pilot submission

DIRAC at KEK

- VMDIRAC 2.0
- CREAM SiteDirector
- Serverless SiteDirector

VCYCLE

- VMDIRAC 2.0
- HTCondor CloudScheduler
- DynamicTorque
- SERVERLESS CE

host cert
proxy cert
host cert
proxy cert
host cert
proxy cert
host cert
proxy cert
host cert
proxy cert
Outline

- Cloud Native applications
- Batch processing in public clouds
- Serverless design in action for the Belle II experiment
- Pricing overhead
What is Cloud Native approach?

- "A native cloud application (NCA) is a program that is designed specifically for a cloud computing architecture. NCAs are designed to take advantage of cloud computing frameworks, which are composed of loosely-coupled cloud services. (...)"

  TechTarget.com

- Cloud Native approach is about building and delivering applications with containers (microservices) and other cloud services, for CI/CD, cloud-native data stores, monitoring and distributed logging systems.

- It fully use distributed nature of the cloud. Such architecture makes the whole system very resilient and quick to deploy (automated DevOps operations).
Decomposition in Cloud Native applications

External service APIs

Functions with the application logic

Internal service APIs
Serverless architecture

Serverless service == zero administration
- We give away provision and manage servers for the cloud provider.
- We give away the OS level configuration and security patches for the cloud provider.
- We give away scaling and fail-over mechanisms for the cloud provider.

Why? Building serverless applications means that your developers can focus on their core product instead of worrying about managing and operating servers or runtimes

AWS Examples
- AWS Lambda, Amazon API Gateway, Amazon S3, Amazon DynamoDB
Serverless batch computation, possible?

The idea was to build a queue system to process jobs with grid authentication - all based on Serverless architecture.
Finally we want to get a REST API endpoint which can handle and process grid pilot jobs, be scalable and maintenance-free.

AWS API Gateway
To provide REST API for HTTP requests handling.
(serverless)

AWS Lambda
To run business logic in Function as a Service approach.
(serverless)

AWS Batch
For batch processing on container clusters. It provides Scheduling and auto-scaling based on the queue occupancy, so we can map one-to-one pilot job and AWS Batch job.
(serverless usage is possible)

Amazon S3
Object storage to store proxies and pilot wrappers temporarily.
(serverless)
Decomposition in Cloud Native applications

External service APIs
CVMFS, other grid services

Functions with the application logic
Job Management, X509 cert validation

Internal service APIs
AWS Batch, AWS S3, AWS CodePipeline
Batch computing services in public clouds

AWS Batch and Azure Batch are quite new solutions providing batch computations based on container clusters.

AWS / Azure Batch as a Serverless service:

- Both providers ensure SLA for more than 99.9% uptime and availability.
- We can specify Docker image managed by CERN.
- Cloud providers offer prepared VM image to work with container clusters.
- Container clusters can scale-out based on jobs in queue.
Interfaces and application logic layer

- Implemented with AWS Lambda and API Gateway.
- Development of Funcion as a Service is not so easy... Lots of communication with other services.
- Authentication based on X509 grid cert for now, but can be extended to other methods like eduGain.
- We deploy Functions and APIs with Terraform tool (Infrastructure as a Code). With this tool you can simulate changes on production environment before applying.
AWS Container cluster system architecture for the Belle II experiment
Scaling AWS Batch cluster based on queues

Scaling depends on amount of jobs to process, limited by wallet thickness. **Scale-out** every few minutes if there are waiting jobs in the queue. **Scale-in** every few minutes if there are not occupied VMs.
AWS Pricing

100 cores for one day

- 35 M4.large (2CPU) * 24h, 70 cores  ~ $85
- EBS Volumes  ~ $4
- EBS Volumes IO  ~ $4
- CloudWatch (optional)  ~ $2
- DataTransfer  ~ $0.7
- Storage  ~ $0.05
- API Gateway  ~ $0.20
- Lambda  ~ $0

AWS Lambda is for free for first million basic request, than $0.20 per million.
API Gateway in our case < $0.20 ($3.50 per million requests).
Total overhead to basic EC2 solution is ~ $1.0
Summary

- Container based batch services are mature enough to manage and scale computing resources automagically.
- Container clusters in AWS and Azure clouds can process Belle II MC production jobs smoothly.
- Serverless architecture ensure a stable service availability like for the LCG Grid computing sites.
- Introduced job management architecture can be used by other grid based projects to easily leverage cloud resources.
- Generic DIRAC's SiteDirector agent has been developed as a client.
- For now we have a full solution for the AWS cloud. Wants to prepare Azure i GCP in future.
Thank you!
Rafal. Grzymkowski@ifj.edu.pl