Service Engineering Research Area
ICCLAB SPLAB
Zurich University of Applied Sciences (CH)

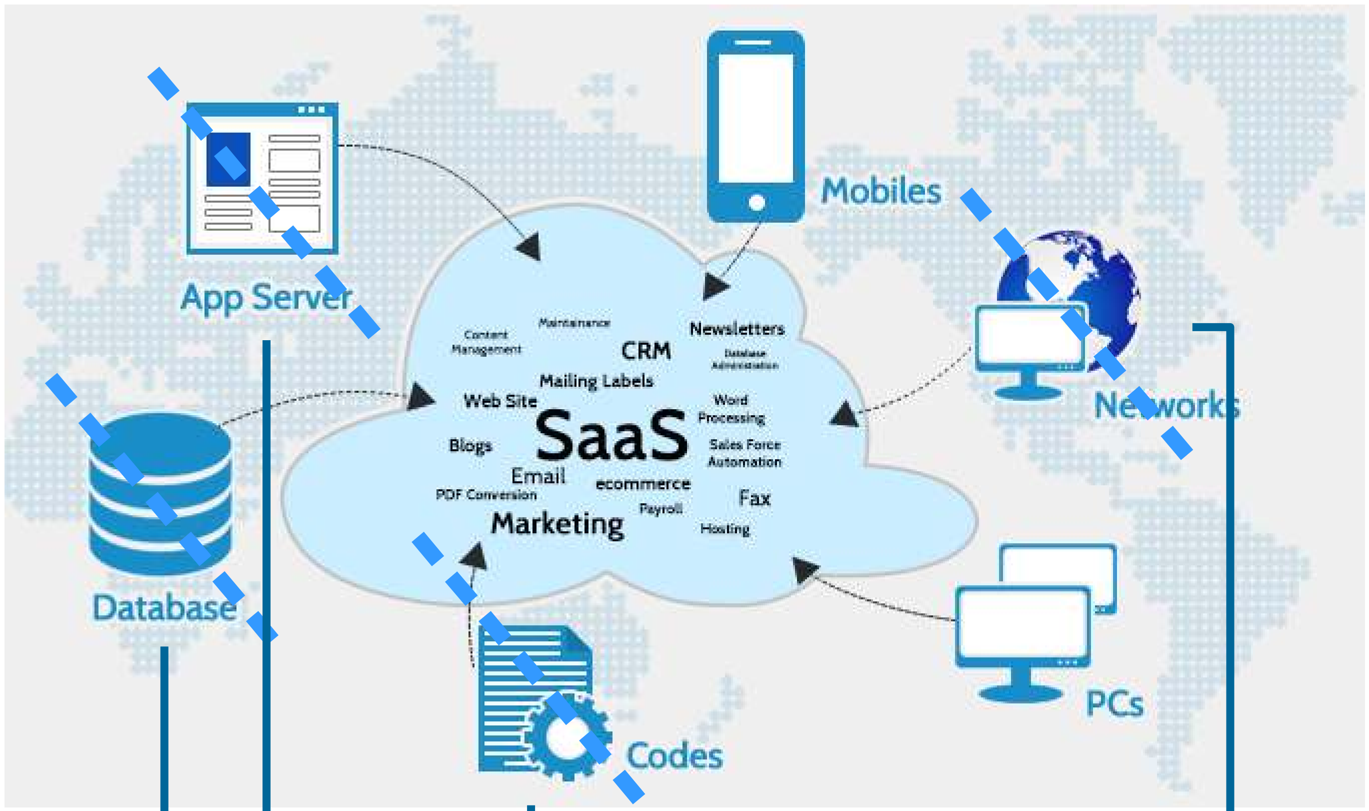
PTI Parque Tecnológico Itaipu
Itaipu Technology Park (PY)

Co-Transformation to Cloud-Native Applications - Development Experiences and Experimental Evaluation

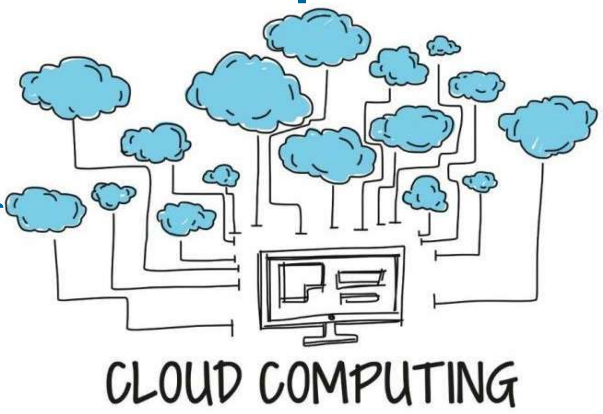
Josef Spillner, Yessica Bogado, Walter Benítez, Fabio López Pires

March 19, 2018 | CLOSER | Madeira, Portugal

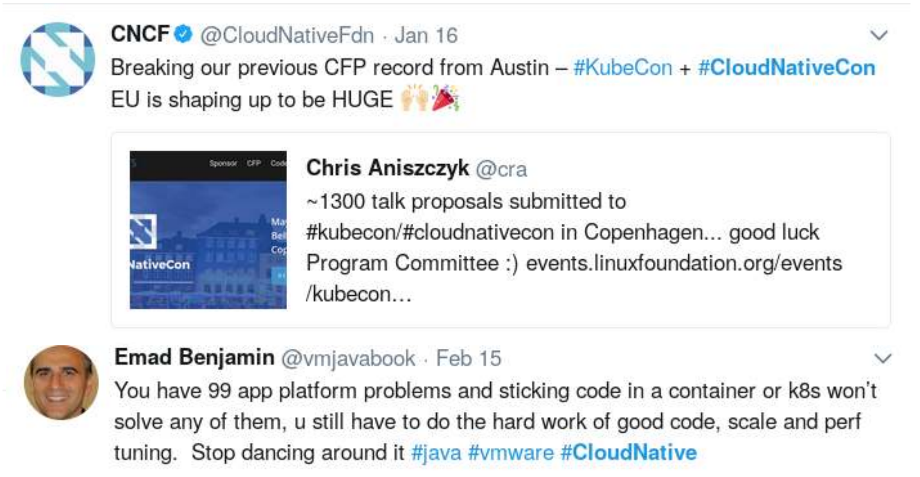
Cloud Applications



[roisinbyrne.co.uk]



Cloud-Native Applications

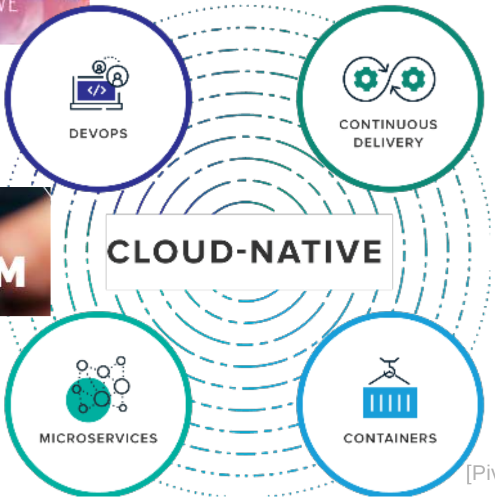


CNCF @CloudNativeFdn · Jan 16
Breaking our previous CFP record from Austin – #KubeCon + #CloudNativeCon EU is shaping up to be HUGE 🙌🎉

Chris Aniszczyk @cra
~1300 talk proposals submitted to #kubecon/#cloudnativecon in Copenhagen... good luck Program Committee :) events.linuxfoundation.org/events /kubecon...

Emad Benjamin @vmjavabook · Feb 15
You have 99 app platform problems and sticking code in a container or k8s won't solve any of them, u still have to do the hard work of good code, scale and perf tuning. Stop dancing around it #java #vmware #CloudNative

[UKCloud]



[Pivotal]

IEEE CLOUD COMPUTING
CLOUD NATIVE APPLICATIONS
SEPTEMBER/OCTOBER 2017
IEEE
acmqueue
Monitoring in a DevOps World
DevOps Metrics
Continuous delivery sounds great, but will it work here?
Containers will not fix your broken culture
Complete table of contents on the following two pages

Cloud-Native Applications

Cloud-Native Computing (CNCF definition 2017):

> ***A new computing paradigm optimised for modern distributed systems environments capable of [ultra] scaling to self-healing multi-tenant nodes.***
>
> **Properties: containerised, dyna-managed, µ-services-oriented**

General views on CNA (de-facto definitions):

Toffetti et. al.	**→ resilient**	
	→ elastic	
ODCA 2015	**→ virtualised**	
	→ loosely coupled	**(composite, discovery)**
	→ abstracted	**(stateless, resilient)**
	→ adaptive	**(0DT live-migration)**

Derived domain-specific views... e.g. for DMS, CRM, ERP, ...

Application Domain: Music Royalties

MRO: Music Royalty Organisations

SUISA GEMA music austria APA PASS MUSICA

MRM: Music Royalty Management

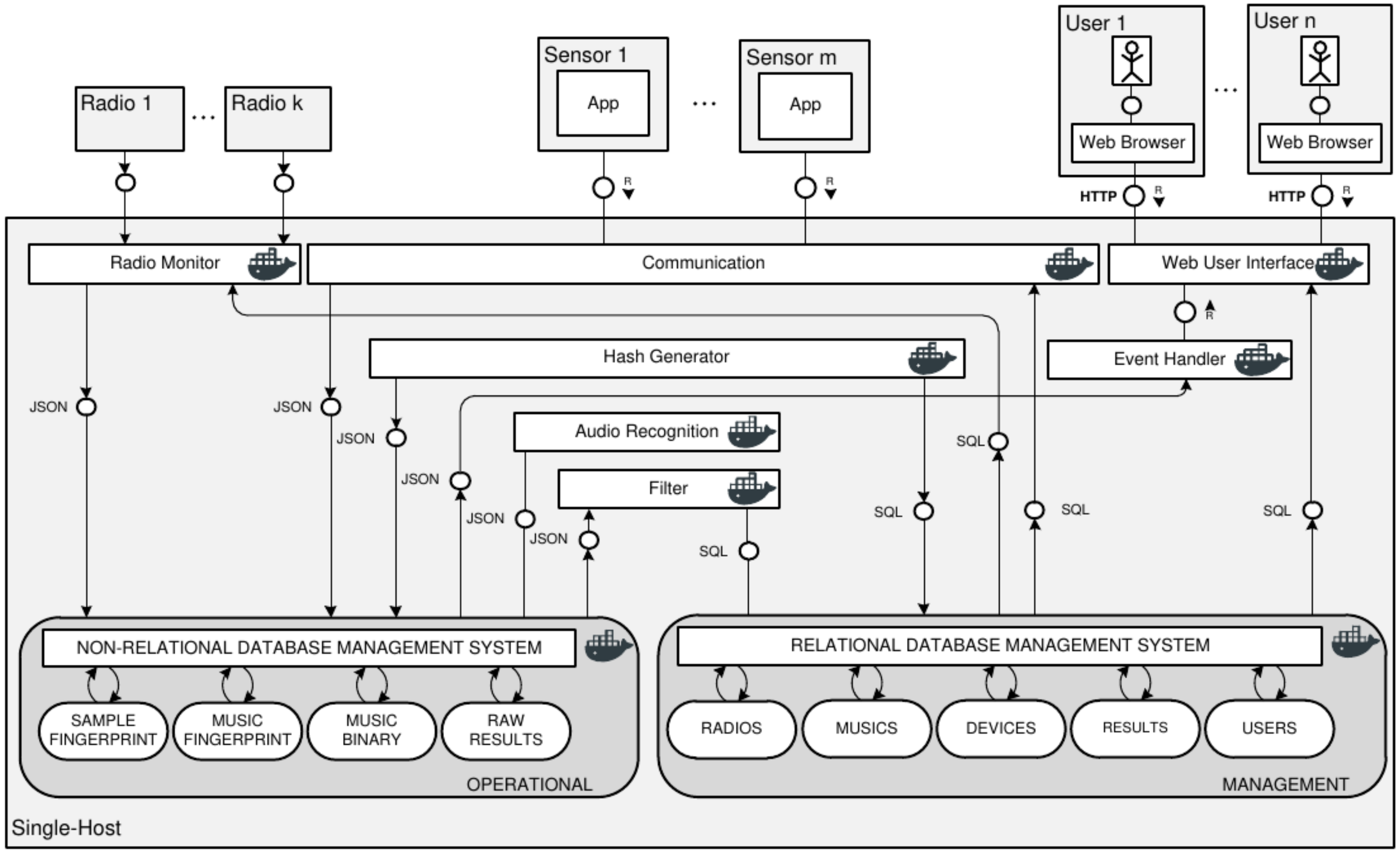
- collection of information about publicly performed works, e.g. music - apart from excempt from royalties
- aggregation and forwarding to MROs

HENDU MRM

- mobile application to detect music played via fingerprint database
- web application for management and bills
- direct access to radio stations

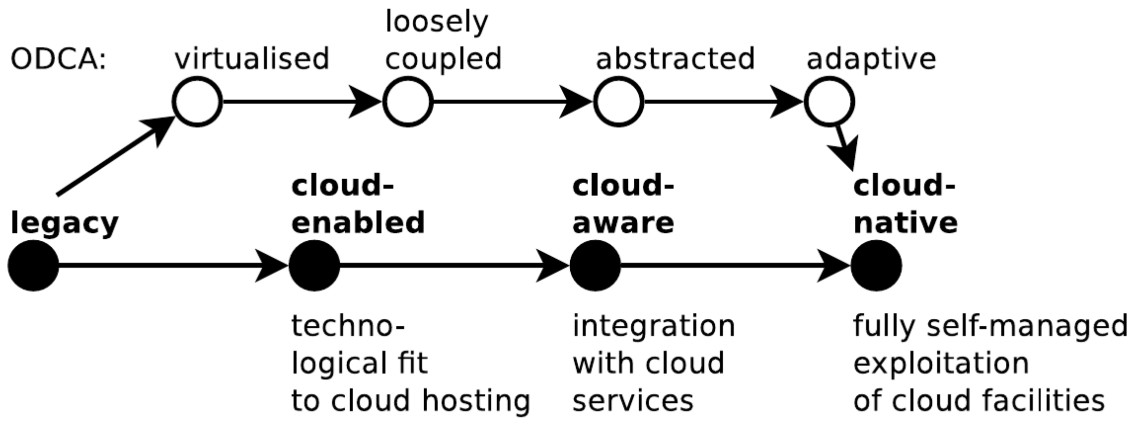
Application Starting Point

Cloud enablement through basic microservices architecture, containers & composite deployment of HENDU



Co-Transformation Methodology

Gradual alignment with highest maturity level (cloud-native)


ODCA:
virtualised
loosely coupled
abstracted
adaptive
legacy
cloud-enabled
cloud-aware
cloud-native
techno-logical fit to cloud hosting
integration with cloud services
fully self-managed exploitation of cloud facilities

From cloud-enabled to cloud-aware:

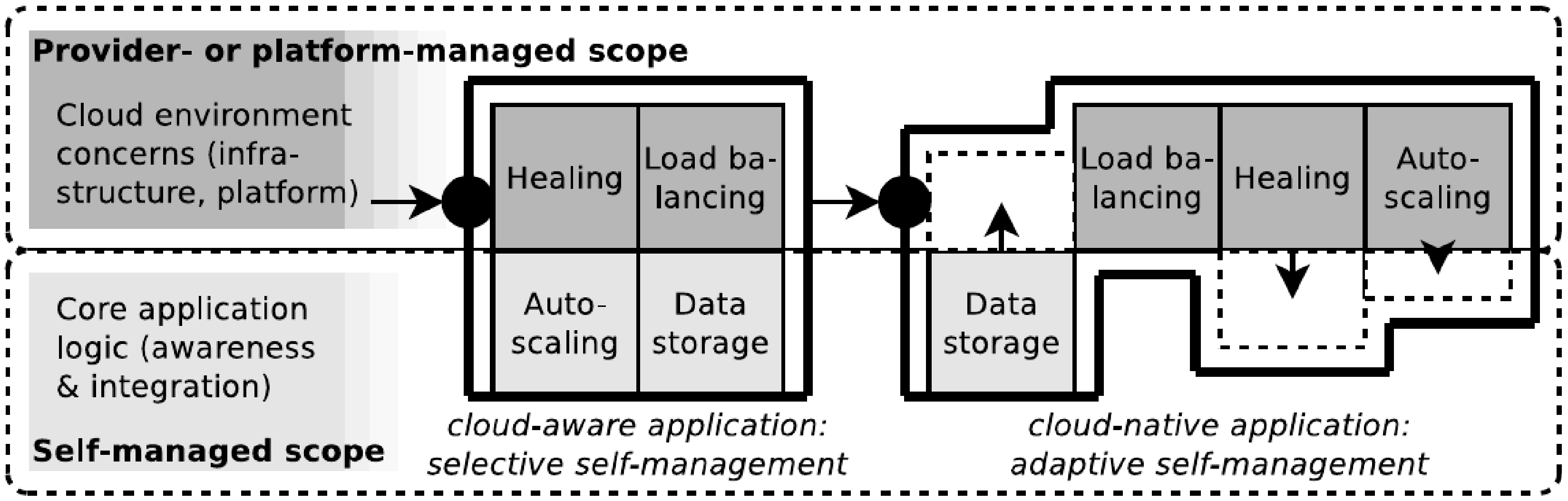
- **discovery & rebinding mechanisms for cloud-provided services**
- **static use of cloud-provided management facilities (e.g. scaling, healing, migration, ...)**

From cloud-aware to cloud-native:

- **separation stateful/stateless microservices + self-management**
- **policies for adaptive enactment of mechanisms**

Self-Management in Detail

Static vs. dynamic (adaptive) choice of management mechanism


Provider- or platform-managed scope
Cloud environment concerns (infra-structure, platform)
Healing
Load ba-lancing
Auto-scaling
Data storage
Load ba-lancing
Healing
Auto-scaling
Data storage
Core application logic (awareness & integration)
Self-managed scope
cloud-aware application: selective self-management
cloud-native application: adaptive self-management

Co-Transformation Steps

Methodology (generic) → Concept (HENDU) → Implementation

From cloud-enabled to cloud-aware:

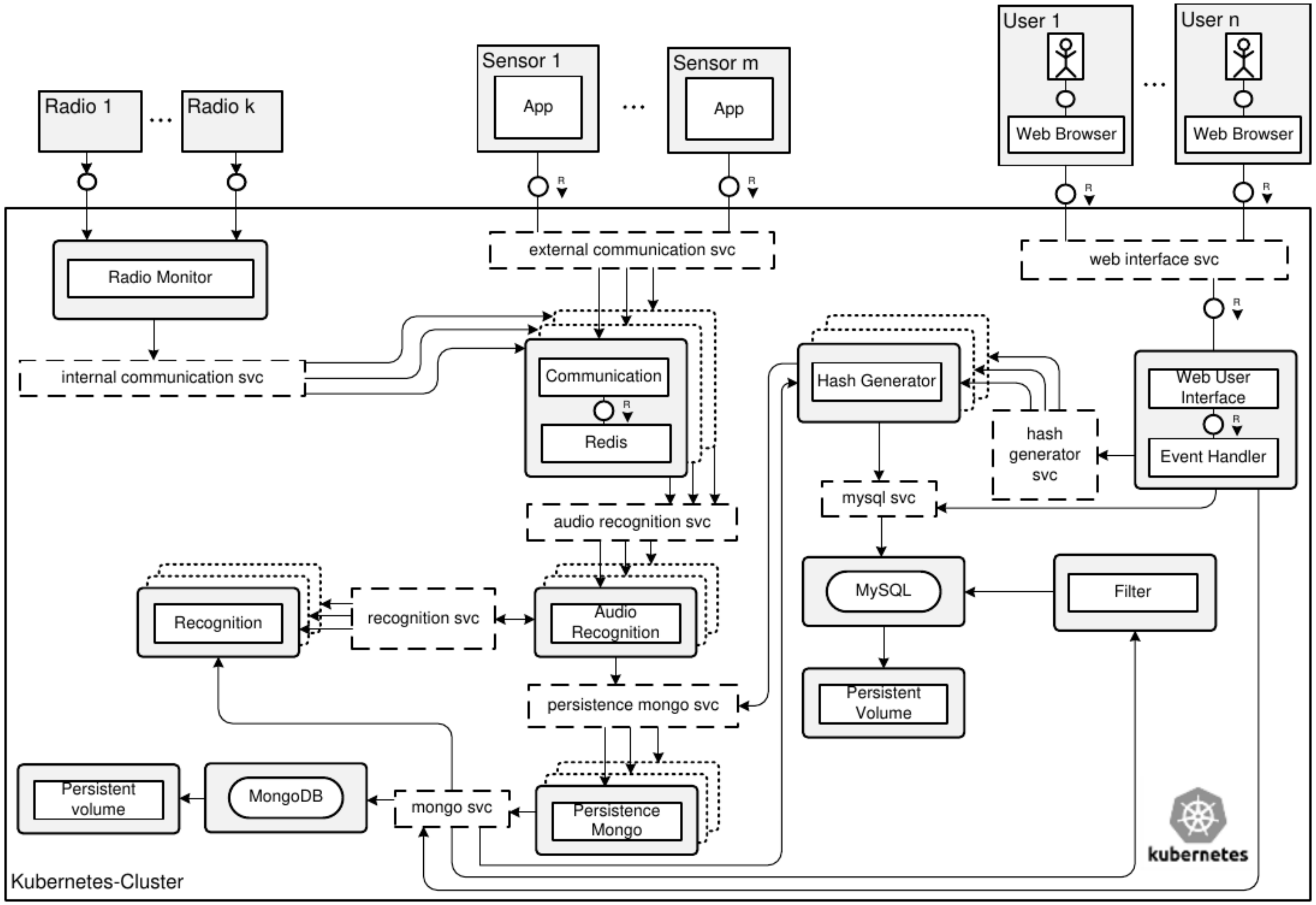
- **flexibility → CA_1 HENDU switch own/platform DBaaS → YAML configuration with endpoints and credentials**
- **platform facilities → CA_2 auto-scaling rules + initial scaling → Kubernetes/Heapster rules based on CPU usage**

From cloud-aware to cloud-native:

- **microservices → CN_1 container images re-engineering → Alpine base images, RESTful endpoints**
- **self-healing → CN_2 health checks → Kubernetes probes**
- **autoscaling → CN_3 domain-specific autoscaling → future work, app-specific metrics**
- **adaptivity → CN_4 application-controlled services and policies → future work, service broker notifications**

Cloud-Native Application Architecture

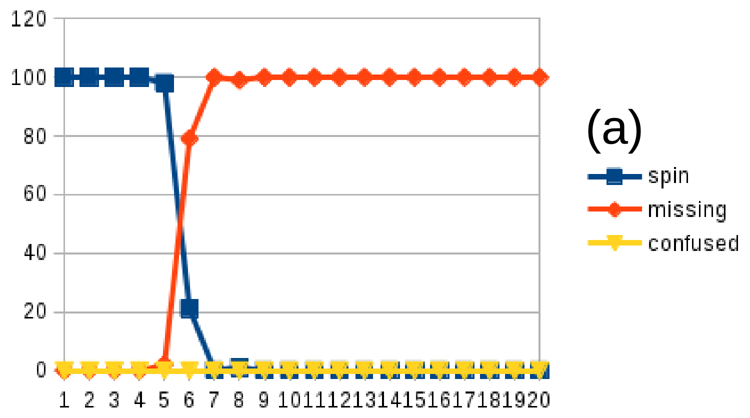
HENDU after successful co-transformation

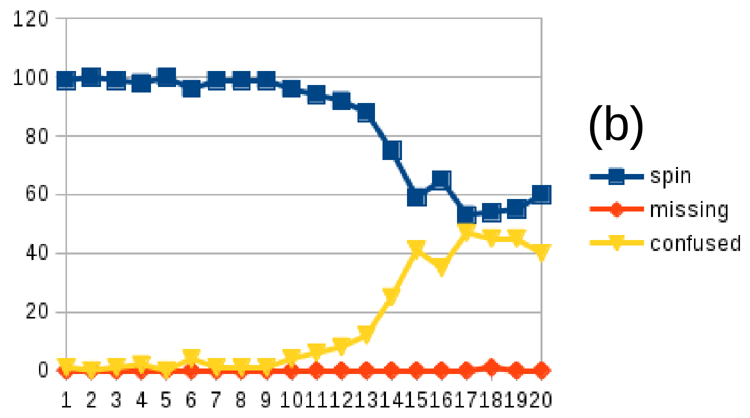


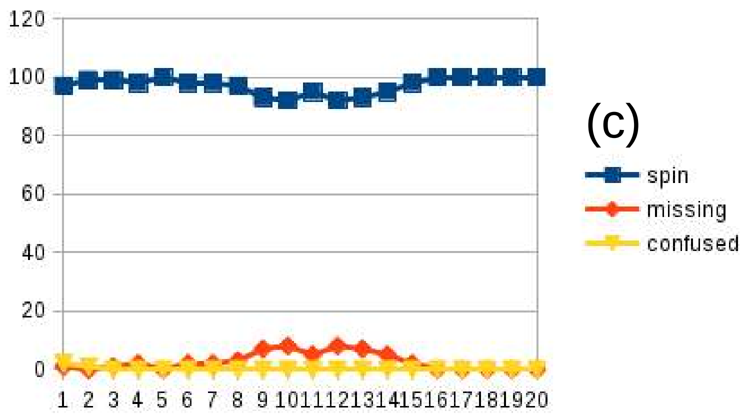
Experimental Evaluation

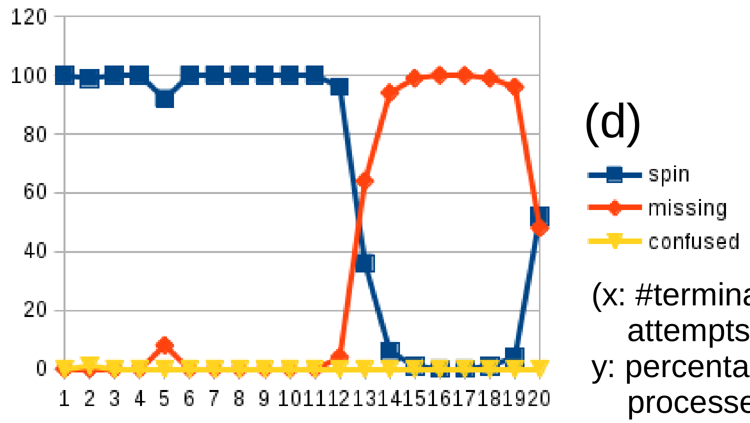
Platform resilience: Docker fault injection + self-healing extension

Experiments: (a) kill containerd-shim, (b) also containerd, (c) «Revive» 5s window when killing containerd-shim, (d) revive 2s









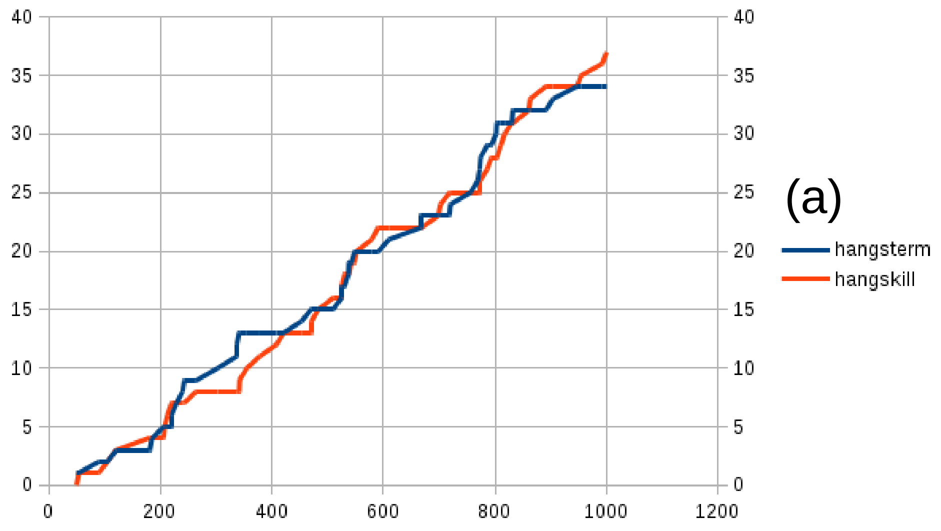
(x: #termination attempts, y: percentage of processes)

Experimental Evaluation

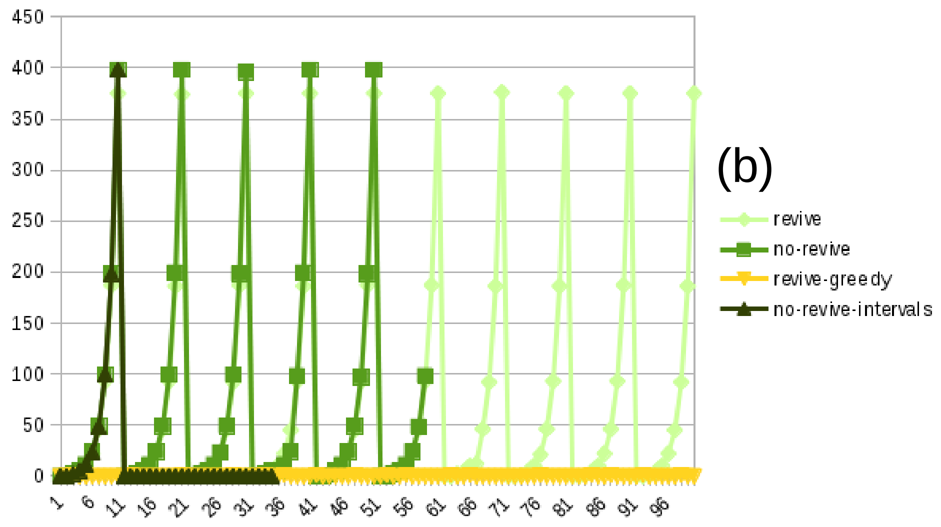
Application resilience with self-healing: «Revive» container

Experiments: (a) 3.55% inconsistent states with SIGTERM/KILL to Docker, (b) exponential backoff with health checks without/with greedy override in «Revive» and 0/1s intervals between signals

Conclusion: immaturity of platforms, also reflects on K8s etc.



(x: #termination attempts, y: inconsistencies)



(x: #termination attempts, y: time in ds)

Experimental Evaluation

Elastic scaling through 2-cluster Kubernetes rules

Workload simulation: JMeter, 100 consecutive HTTP POSTs, 10min

Results:

→ with CPU auto-scaling:	**RT=120ms**	**RS/m=2833**
→ without CPU auto-scaling:	**RT=2923ms**	**RS/m=1213**

Conclusion: good non-linear scalability of platforms

Conclusions

Already achieved

- First systematic cloud-native transformation approach for software developers
- Number of tools such as docker-killer (published through OSF)

Still to be worked on

- Complete self-management including cross-provider migration
- More fine-grained workflow with serial and parallel steps
- Automation tools for future co-transformations
- Application of methodology in other domains

Follow our work

- Cloud-Native Applications research initiative @ SPLab (since 2014)