Checkpoint/Restore v kontajneroch

Na hrane zajtrajška

Viktória Spišaková & Lukáš Hejtmánek, 13.10.2024

Kto sme?

A čo robíme?

- Centrum CERIT-SC v rámci e-INFRA CZ
- Posledných pár rokov (2-3) budujeme Kubernetes ako platformu pre beh kontajnerov pre potreby členov e-INFRA CZ
  - Dokopy[1] 3456 CPU, 13TiB RAM, 39 NVIDIA GPU
- A prečo?
  - Kontajnere inherentne zabezpečujú reproducibilitu (podstatné pre vedu)
  - Defacto štandard poskytovania zdrojov v cloude
  - Cloud poskytuje elasticitu

1) https://docs-ng.cerit.io/en/platform/hw, dokumentácia je WIP

Pre koho sú Kubernety vhodné?

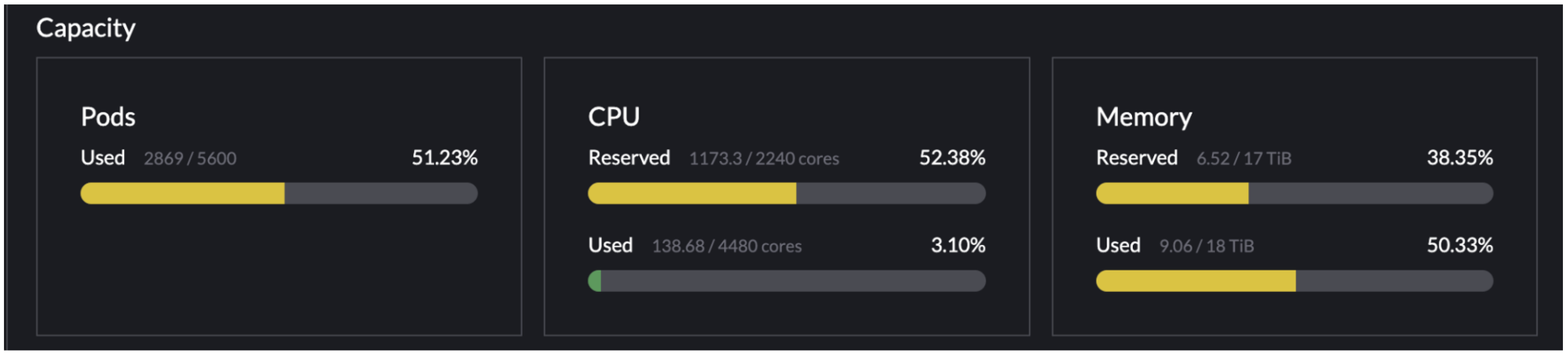
Vítame všetkých.

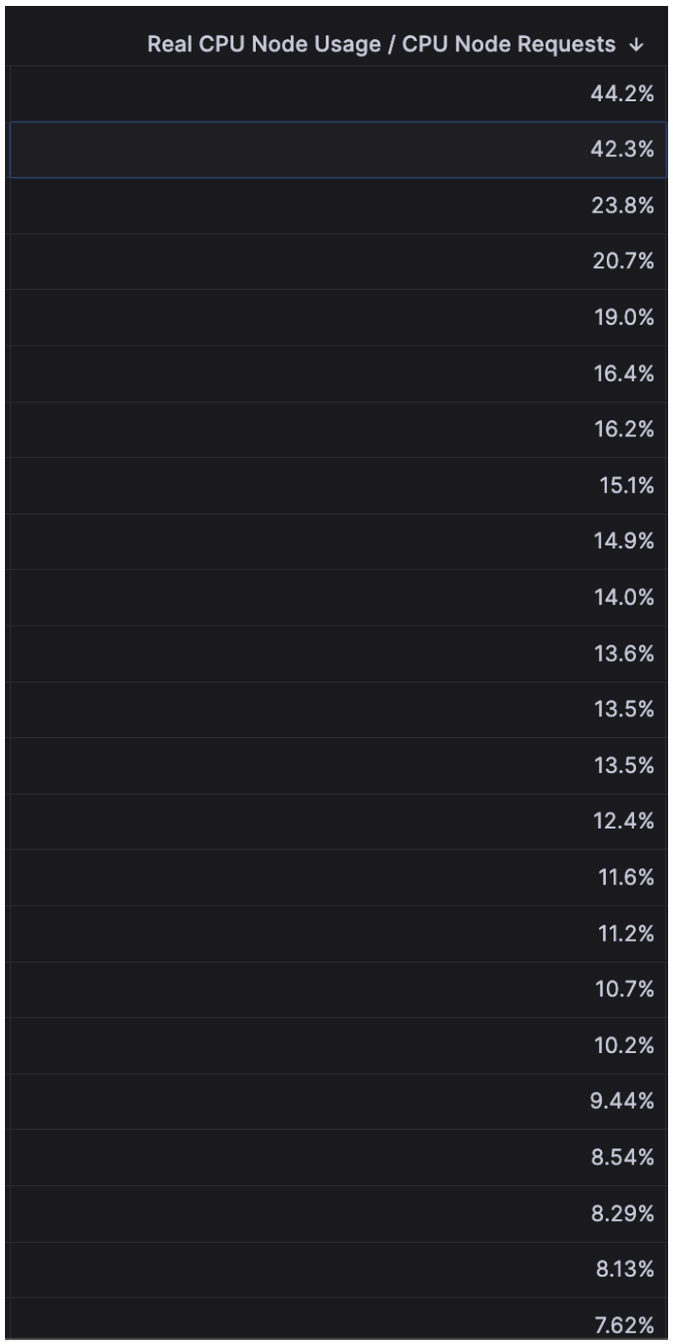
- Statické aplikácie (weby), runnery
- HPC workflowy - bioinformatické pipelines
- Interaktívne workloady
  - Web-based computing (JupyterHuby), virtuálne desktopy, grafické aplikácie
- Náročné výpočty (SW pre kryoelektronovú mikroskopiu)
- Dlhotrvajúce výpočty (predikcie proteínov)
- Bursty workloads

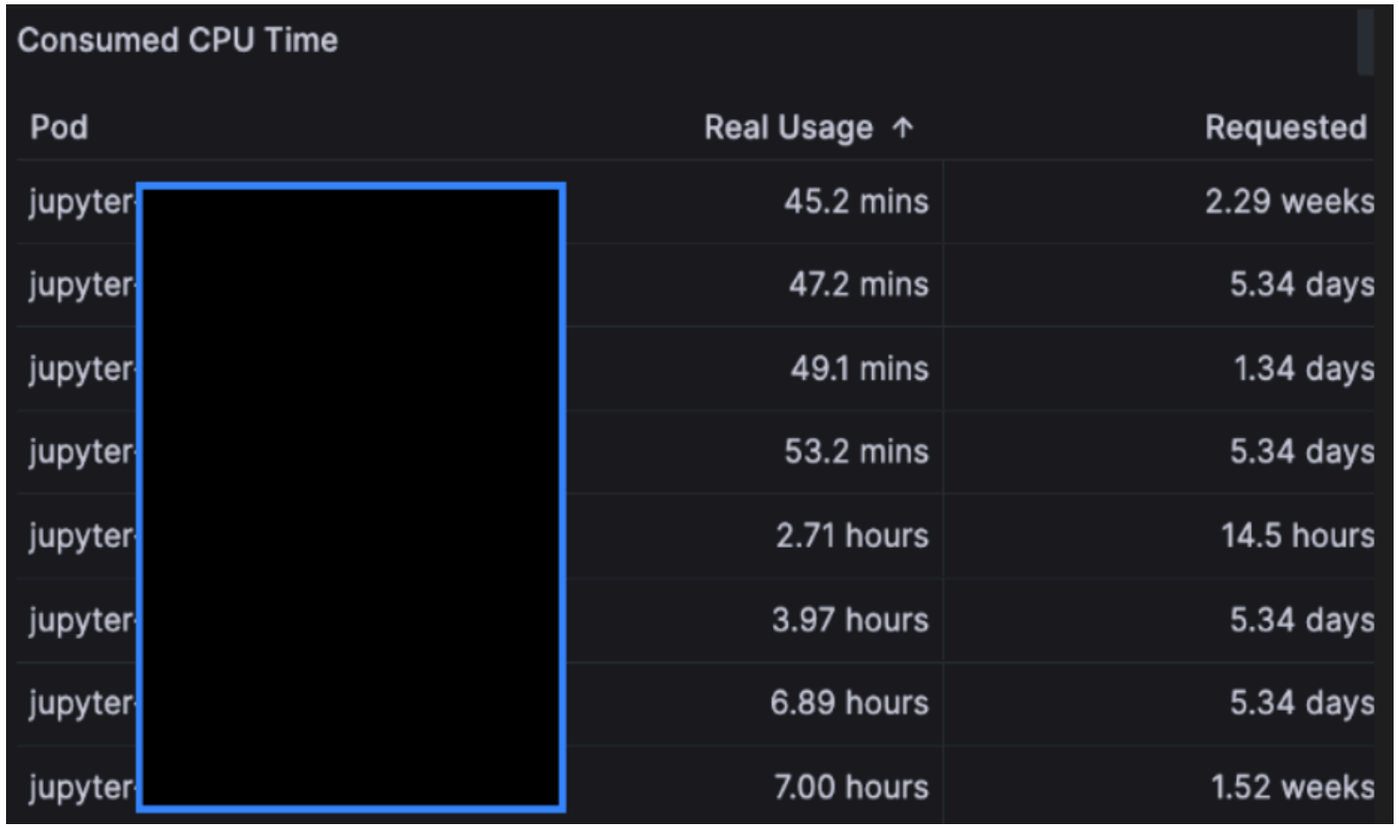
boj o zdroje medzi všetkými

Takže máte vyťažený cluster?

... 😶





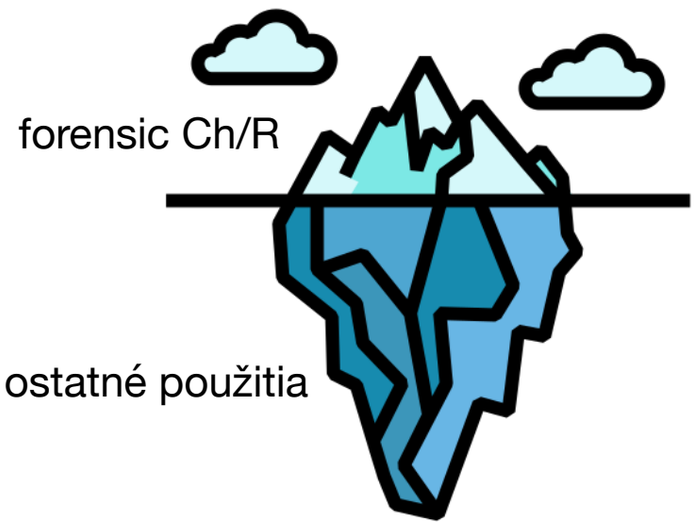


Transparentný Checkpoint/Restore.

- Checkpoint/Restore umožňuje uložiť stav kontajneru na disk a neskôr spustiť od toho miesta
- Transparentný = funguje na každý kontajner bez rozdielu a bez jeho asistencie
  - Kedže ide o parazitický kód
- Na nízkej úrovni zabezpečuje Linuxová utilitka CRIU, ktorá je ďalej podporovaná v container runtimes a *nejak* v Kubernetoch
  - V Kubernetoch ako forensic checkpoint/restore čo nie je veľmi zaujímavé

Zaujímavejšie použitia.

Sú všetky ostatné.



- Živá migrácia
  - Uzol do maintenance? Zmigrovať bez prerušenia
  - AWS Spot Instances 2m notice? Zmigrovať bez prerušenia
- Fault tolerance a HA
  - Checkpointovanie ML tréningov[1]
  - Rýchlejšia recovery po node failure
  - Back-up dlhotrvajúcich úloh
  - Keby sme vedeli urobiť checkpoint tesne pred OOM …

1) https://www.usenix.org/conference/fast21/presentation/mohan, https://dl.acm.org/doi/10.1145/3477314.3507221, https://dl.acm.org/doi/10.1145/3600006.3613145

Zaujímavejšie použitia.

Sú všetky ostatné.

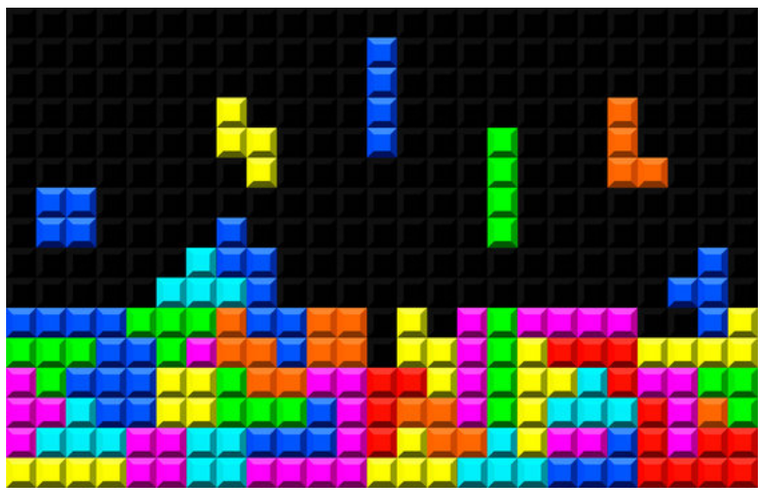
- Plánovanie na steroidoch
  - Oportunisticky alokovať zdroje a checkpointovať keď zdroje sú potrebné pre niečo iné
    - Úloha sa checkpointne ak do systému príde nová, vyššie prioritná úloha
    - Checkpoint úlohy a restore s inými zdrojmi
- Prostredia pre debuggovanie (forenzná analýza)
- BTW: GPU checkpoint/restore funguje (AMD, NVIDIA malé fixy)

1) https://www.usenix.org/conference/fast21/presentation/mohan, https://dl.acm.org/doi/10.1145/3477314.3507221, https://dl.acm.org/doi/10.1145/3600006.3613145

Benefity Checkpoint/Restore.

Pre infraštruktúru.

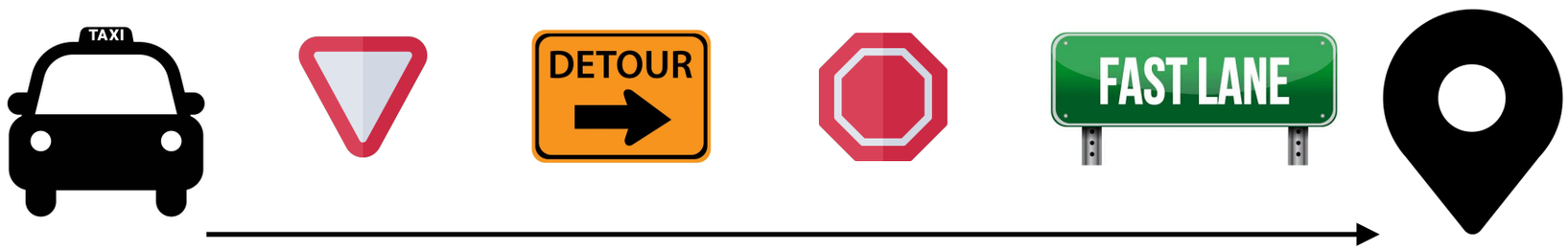
- Hustejšie zaplňovanie fyzických zdrojov bez nutnosti natvrdo zabiť úlohu pri blížiacom sa overloade
    - Kill = dvojmo prepočítaný čas (a raz stratený) pretože úloha sa musí spustiť zas od začiatku
- Underload = restore už checkpointovaných úloh alebo spustenie nových
- Ch/R umožňuje +- in-real-time meniť zloženie bežiacich workloadov podľa priorít, aktuálnej potreby alebo typu workloadu



Benefity Checkpoint/Restore.

Pre užívateľa infraštruktúry.

- On-demand alokácia viac zdrojov pre výpočty (ak sú dostupné)
  - Užívateľ musí počítať s prerušením
  - Time-to-result teda nemusí byť rovnaký, čo čisté trvanie úlohy keď sa pustí na rezervovaných zdrojoch
    - Na druhú stranu, rezervácia na dedikovanom stroji môže nastať až o niekoľko dní a teda beh hoc i s prerušeniami môže byť rýchlejší
- Checkpoint zadarmo poskytuje ukladanie “medzivýsledkov” a nie je potrebné programovať vlastný mechanizmus pre prípad chyby či výpadku (aplikačný ch/r)



Zhrnutie.

- Checkpoint/Restore je zaujímavá utilitka ako pre infraštruktúru, tak pre užívateľov

- Užívatelia budú checkpoint/restore používať, ak ich nebude stáť žiadnu energiu a prinesie im výhody za cenu “malých nevýhod”

  - Podobné myšlienke FaaS — nemusíte sa starať o infra, upriamte sa na svoj kód

- Infraštruktúry budú checkpoint/restore používať pretože im umožní efektívnejšie využívať výpočetné kapacity a zároveň pružnejšie reagovať na aktuálne potreby

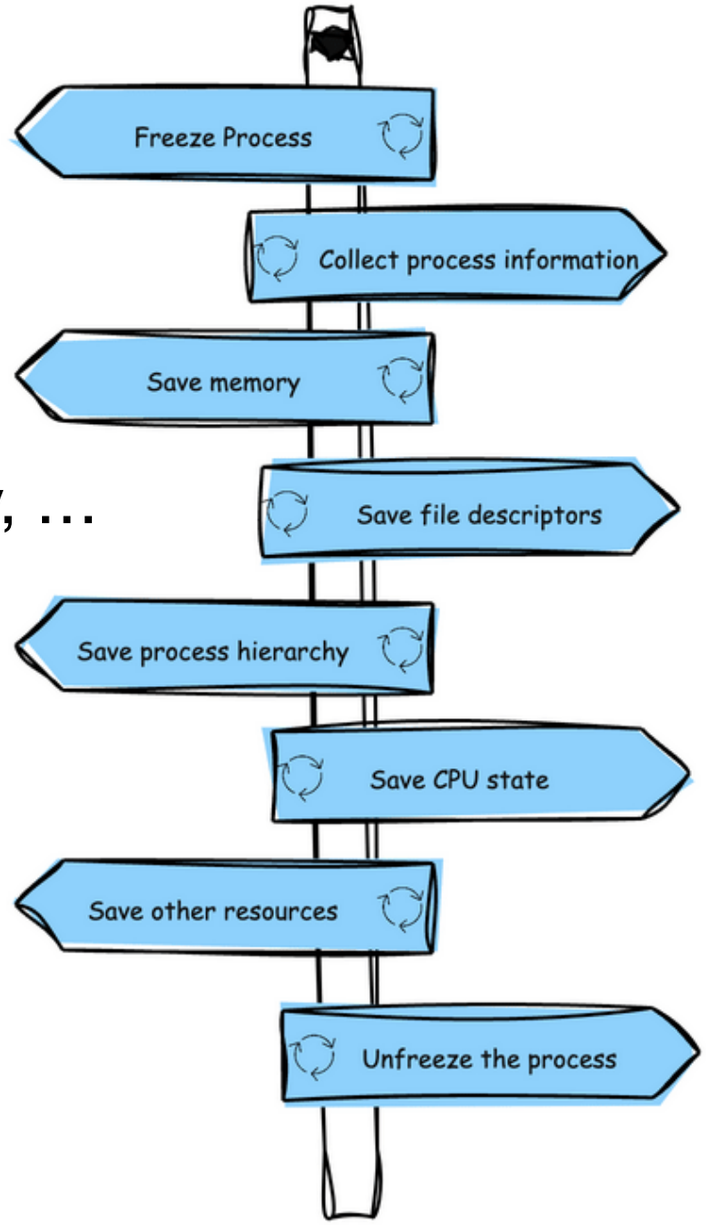
  - Môže vzniknúť nová trieda kvality workloadov

Každá složitější technologie je nerozeznatelná od magie

Technická realizace

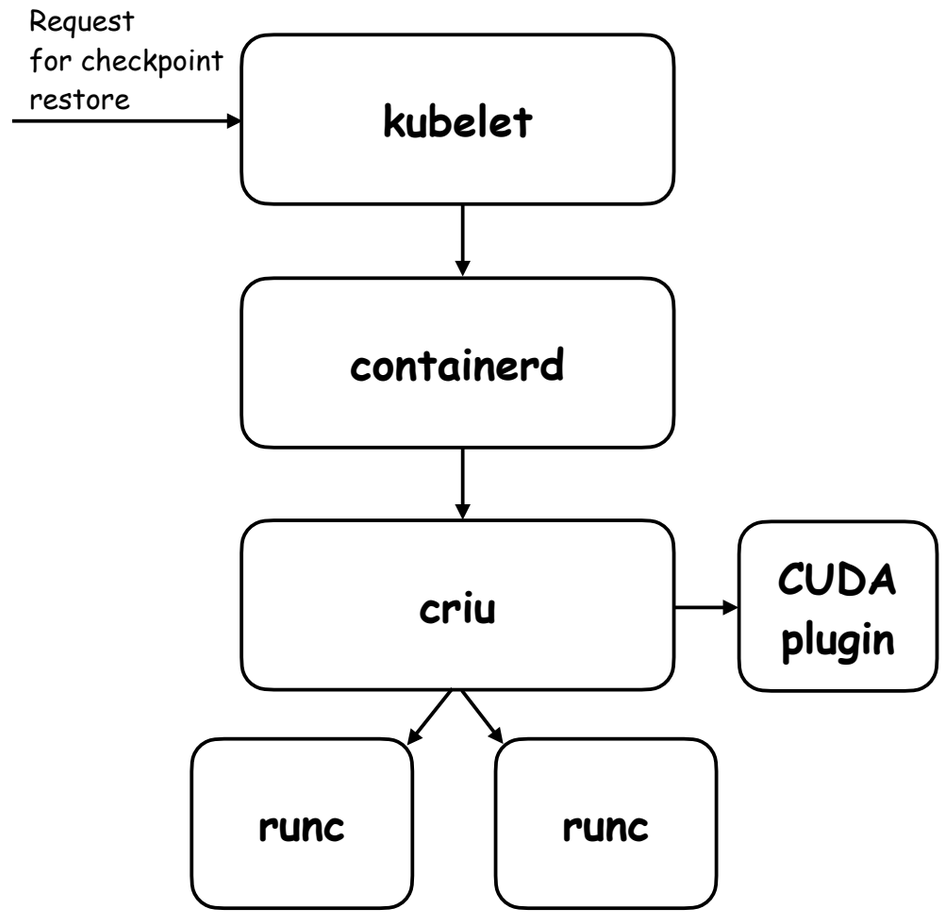
Petrificus Totalus

- Runtime: Podman, Containerd (K8s)
- CRIU
  - Userspace nástroj pro výrobu checkpointu
    - Stav CPU, obraz paměti, otevřené soubory, zámky, …
    - Pluginy např. pro CUDA/GPU
- Checkpoint
  - Vyrobí TAR
- Restore
  - TAR nebo docker image



Technická realizace v K8s

- Featuregates: ContainerCheckpoint=true
- Containerd ve verzi 2.0
- runc ve verzi 1.2
- criu 3.19 nebo lepší
- Checkpoint
  - curl volání na kubelet API
- Restore
  - vytvoření nového Podu se speciálním docker image




LIVE. DIE. REPEAT.
EDGE of TOMORROW

Na hraně zítřka

Co všechno (ne)funguje

- Uvažovaný runtime
  - Podman, Containerd, CRI-O

- Podman
  - Checkpoint CPU/Paměť ✅
  - GPU ✅
  - Síť ✅
  - Speciální zařízení ❌

- Containerd, CRI-O
  - Checkpoint CPU/Paměť ✅
  - GPU ❌
  - Síť ✅
  - Speciální zařízení ❌

Na hraně zítřka

Co všechno (ne)funguje v K8s

- Runtime Containerd
- Chybí podpora v API
  - Pro Checkpoint a hlavně Restore
- Problematická síť
- Speciální HW (vč. GPU) nefunguje
- Šifrování
- Násobné kopírování dat
- Interakce s vnějším světem
- A vše ostatní vlastně jde už teď 😎👉

Ch/R v K8s

Nemožnosti prvního řádu

- Nefunkční GPU
  - CRIU s CUDA pluginem — Checkpoint funguje (Vyjma UVM)
  - Restore má zatím drobné problémy (na úrovni mountu runtime)
- Šifrování checkpointu — potenciálně citlivá data — env
  - Prototyp od Radostina Stoyanova funkční

Ch/R v K8s

Nemožnosti druhého řádu
aneb nejvíce problémů si lidé způsobují sami

- Chybějící podpora API
    - Složité prosazování změn do celého systému Kubernetes
    - Hlavní Kube API nemá podporu Ch/R (byť ma featuregate)
    - Podpora v API komponenty Kubelet
- Restore není vůbec vymyšlený
    - Aktuální přístup: vyrobení nového Podu

Ch/R v K8s

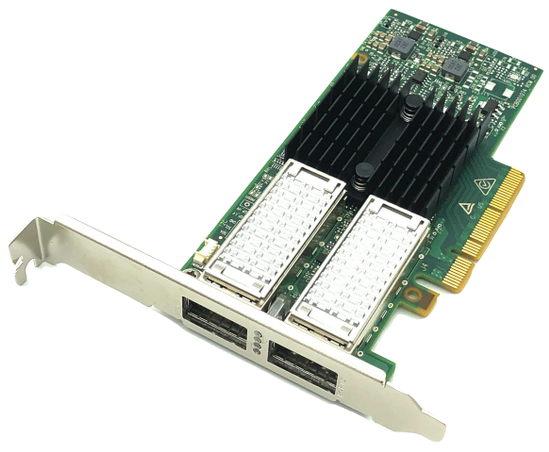
Nemožnosti druhého řádu

- Násobné kopírování dat
  - kubelet vyrobí tar
  - containerd má dočasně od tohoto taru docker image
  - pro restore je z taru vyroben nový docker image
  - nahraje se do registru
  - stáhne se z registru
  - rozbalí
  - spustí

Ch/R v K8s

Nemožnosti třetího řádu

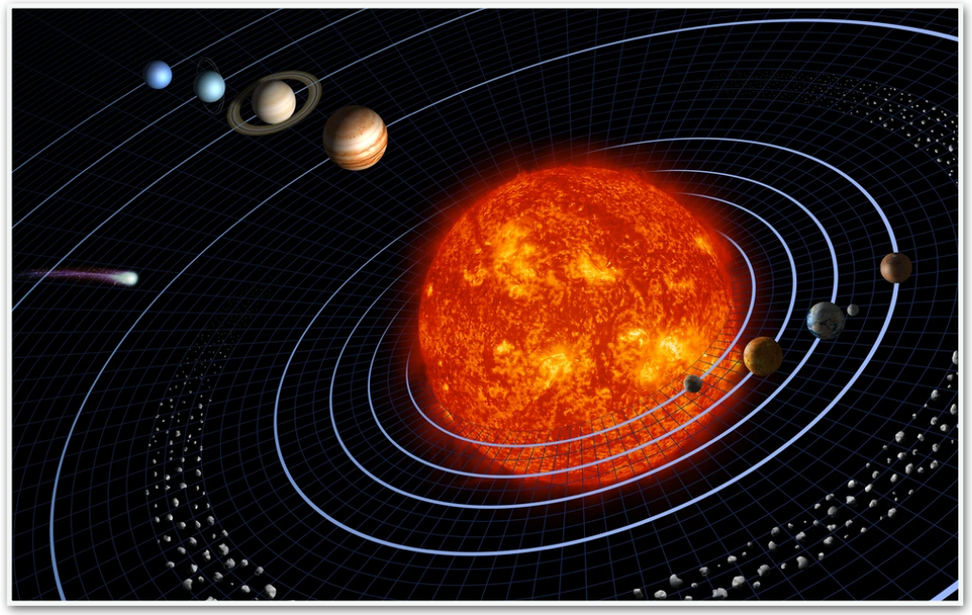
- Speciální HW (vyjma AMD/NVIDIA GPU)
  - Neexistuje obecný mechanismus á-la suspend/resume v kernelu
    - Někdo musí naprogramovat uložení stavu HW
      - blízké suspend to RAM
      - po checkpointu se může pokračovat v běhu



Ch/R v K8s

Nemožnosti třetího řádu

- Obnova síťového spojení
  - CRIU umí obnovit TCP stack
  - Smůla, že v K8s standardně nelze obnovit IP adresu
    - Každý uzel má vlastní C podsíť
      - Teoreticky lze změnit, ale …
- Infiniband
  - Obnova sítě extra složitá
    - Globální GUID na stroji
    - Ch/R jen některého kontejneru



Ch/R v K8s

Nemožnosti čtvrtého řádu aneb perpetum mobile

- Interakce s externím světem
  - Externí úložiště dat
    - Potřeba synchronního snapshotu — ne vždy je vůbec realizovatelné
  - Externí API
    - Jak vrátit i jeho stav do bodu checkpointu?

Je to opravdu tak zlé?

- Není!
- Koncepční implementace funguje
- Pokusy na aplikacích ukazují funkčnost
- Pro většinu zmíněných problémů existuje řešení
  - Některé budou trvat
- Daří se pomalu přesvědčovat komunitu, že je Ch/R užitečný

Jupyter Notebook Checkpoint/Restore

Demo

- https://drive.google.com/file/d/1Q3UWJWbkAvYFQVtkCKLS7K8HHlS6z9QV/view

LLM Checkpoint/Restore

Demo

- https://drive.google.com/file/d/1x1a8va00isedBeKNMnlGVKDKfUD50Qno/view
- Video: Radostin Stoyanov, University of Oxford
- Druhá polovica videa demonštruje encrypted checkpoint/restore

Děkujeme za pozornost.

k8s@ics.muni.cz