

ОПТИМИЗИРАНЕ НА КОНТЕЙНЕРИ ЗА ПАРАЛЕЛЕН КЛЪСТЕР

Иво Илиев¹, Ана Пройкова^{1,2}

¹Лаборатория HPC в Лабораторния комплекс на СТП,
hpc-lab@sofiatech.bg

²Център за върхови постижения в ИКТ “УНИТе”, Софийски университет
„Св. Кл. Охридски“
anar@phys.uni-sofia.bg

Резюме: Контейнеризацията е много полезна софтуерна концепция, която се оказва нетривиално приложима в среда HPC (High Performance Computing). Контейнерите се основават на съществуващите функции на ядрото и позволяват по-добър потребителски контрол за тяхното приложение. Основният проблем, който трябва да се реши при контейнеризиране в HPC, е евентуалната загуба на производителност. Анализирайки проблема, открихме, че архитектурата на контейнера Singularity превъзхожда контейнера Docker. Работата е извършена по отношение на хомогенен изчислителен клъстер Nestum <http://hpc-lab.sofiatech.bg/>. Резултатите обаче остават валидни за широк клас HPC системи.

Ключови думи: HPC, supercomputers, containers, numerical & computation science

1. Въведение

Контейнеризацията е изключително полезна софтуерна концепция, която обаче се оказва нетривиална за имплементиране в условията на HPC (HPC – high performance computing, високопроизводителни изчисления). Контейнерите се основават на вече съществуващи свойства на ядрото на операционната система и позволяват за по-голям контрол от потребителя на запусканите програми – с какво те могат да си взаимодействат и като цяло какво за тях е видимо.

2. Изследване на контейнери в паралелен изчислителен клъстер

За HPC имплементации това обикновено е рестриктирано до пространството `namespaces mount`.

Nestum използва софтуера Slurm (де-факто стандарта за HPC средите) за управление (менажиране) на задачи и натоварване. Тоест, преди всичко трябва да е ясно дали контейнерния софтуер, който искаме да ползваме, е съвместим със Slurm. За щастие Slurm е „агностичен“ от към контейнери и може да работи и стартира почти всички видове. Следва списък на някои контейнерни софтуери, които имат работещи плъгини за Slurm:

- Charliecloud
- Docker
- UDOCKER
- Kubernetes Pods (k8s)
- Shifter
- Singularity
- ENROOT
- Podman
- Sarus

2.1. DOCKER, uDOCKER, SINGULARITY

Макар популярен, **Docker**, има няколко проблема, които го правят неподходящ за имплементация в HPC системи. Основният проблем е, че по дефиниция само привилегировани потребители трябва да имат право да контролират Docker daemon-а. Това като цяло е недопустимо при повечето HPC реализации. С други думи, на практическо ниво, Docker дава superuser привилегии, което прави трудно ограничаването на един потребител от това да има достъп до данните на друг.

Допълнително, инфраструктурата на Docker е такава, че Docker не може по тривиален начин да изиска ресурси от Slurm.

uDocker: Този софтуер клонира подмножество от функции на Docker и позволява работата им изцяло в пространството на потребителите (user space). uDocker също позволява модификация на вече съществуващи Docker контейнери, стига модификациите да включват инсталиране и модифициране на файлове само вътре в контейнера.

Добра практика е uDocker да се използва за запускане на предварително подготвени Docker контейнери.

uDocker е наличен на Nestum посредством модула си, *module load udocker*.

Singularity: Тази хибридна контейнерна система започва бързо да се налага като предпочитаното решение за HPC системи. Основните плюсове на Singularity е, че поддържа:

Интеграция със Slurm посредством добре разработен plugin. Singularity изпълнява контейнерите като нативни (native) програми или скриптове на host компютъра, което позволява възможно най-пълна интеграция със Slurm (или друг scheduler).

Контейнери в потребителско пространство (user namespace) посредством режим ‘пясъчник’ (sandbox mode), които не изискват допълнителни права.

Стандартният вход, изход, грешка, тръби и други комуникационни канали, които се използват от локалните програми са синхронизирани с приложенията, вървящи локално в контейнера.

Възможност потребителите да извикват singularity посредством `setuid` извън Slurm.

Използването на `setuid` е изключително полезно – това е стар и изпробван Unix метод и води до гъвкавост в термини на поддържани функции и поддръжка на legacy софтуер.

Singularity е инсталиран и достъпен контейнер за всички потребители на Nestum. На адрес <http://hpc-lab.sofiatech.bg/containers/> потребителите могат да намерят примерни конфигурации и насоки за използване на софтуера. Допълнително, на Nestum фигурират няколко пре-конфигурирани контейнерни изображения, готови за използване – едно, съдържащо Gromacs 2020.1 и едно - съдържащо Tensorflow.

От основна полза за хората използващи научен софтуер е добрата интеграция на Singularity с Message Passing Interface (MPI), което е стандартен API използван от много научни пакети и софтуери. Singularity е out-of-the-box съвместимо с OpenMPI(v2). Интеграцията е скрита от потребителя – от негова гледна точка е нужно просто да се извика `mpirun`, все едно се намира на host компютъра. Допълнително, не се наблюдава значима загуба на производителност, ако се изпълняват OpenMPI задачи във Singularity контейнер.

Singularity е построено, така че контейнерното изображение е напълно преносимо – не е нужна система, подобна на слоевете при Docker. Тестове на Singularity containers на голям брой изчислителни конфигурации (не свързани с нашата лаборатория) показват, че изпълнението на задачи от Singularity контейнери не води до осезаеми загуби на производителност (дори и при паралелни задачи върху възли, използващи Infiniband).

Заклучение

Въз основа на тестовете с приложенията GROMACS и Tensorflow и обобщавайки коя рамка е подходяща за потребителите на Nestum може да се посочи, че Singularity е ясният победител и трябва да бъде препоръчан като рамка за контейнеризация, а също така и за приложение в облачен HPC.

Благодарности

Авторите благодарят за използването на HPC клъстера NESTUM по договор със СНИРД-СТП, № Д-140-2019/03.12.2019 г. и изразяват благодарност за частичната финансова подкрепа от Националната научна програма Информационните и комуникационни технологии в науката, образованието и сигурността (ИКТ в НОС), финансирана от Министерството на образованието и науката.

Литература

1. <http://hpc-lab.sofiatech.bg/containers/>
2. <http://www.gromacs.org/>
3. <https://slurm.schedmd.com/documentation.html>

OPTIMIZATION OF CONTAINERS FOR THE HPC CLUSTER „NESTUM“

Ivo Iliev¹, Ana Proykova^{1,2}

¹ HPC Laboratory in the Laboratory Complex at the Sofia Tech Park, hpc-lab@sofiatech.bg
² Centre of Excellence in ICT “UNITE” at Sofia University, anap@phys.uni-sofia.bg

Abstract: Containerization is a very useful software concept, which turns out to not be trivially implementable in HPC (High Performance Computing) environments. Containers are based on existing kernel features and allow for better user control for their application. The main problem that arises when considering containerization in HPC is a loss of performance. Having that in mind, we found the Singularity container architecture is outperforming the Docker container. Even though the work was done in relation to the cluster Nestum <http://hpc-lab.sofiatech.bg/>, they remain valid for a broad class of HPC systems.