

Migracja maszyn wirtualnych, procesów i kontenerów w chmurach

Grzegorz Dwornicki

Wydział Zastosowań Informatyki i Matematyki SGGW

Coraz większą popularnością cieszy się cloud computing nazywany potocznie „chmurą”. Chmura w wersji „IaaS”, to usługa wypożyczania bezpośredniego zasobów sprzętowych, a więc czasu procesora, pamięci RAM, pojemności dysków itp. W przypadku innych typów chmury („PaaS”, „SaaS”) wypożyczanie zasobów sprzętowych jest pośrednie, czyli nie widoczne bezpośrednio dla użytkownika. Usługa chmury komputerowej jest w swoim założeniu wysoko dostępna i wysoko niezawodna stąd wynika konieczność migracji usług na inny serwer w chmurze w przypadku zbyt dużego obciążenia lub awarii pierwszego. Podczas pracy klient łączy się z aplikacją w takiej chmurze i komunikując się z nią zleca jej zadania lub odbiera wyniki pracy. Aplikacja może zostać przemieńczona na inny serwer podczas swojej pracy. Na seminarium postaramy się odpowiedzieć na nurtujące wszystkich użytkowników chmur pytania. Czy klient może coś powiedzieć o tej migracji? Czy może ją zauważyć? Czy aplikacja może zaobserwować skutki swojej migracji?

Początkowo mówiło się o chmurach używających maszyn wirtualnych jako pojemników na usługi, a od kilku ostatnich lat mówi się o chmurze opartej na tzw. kontenerach [1-5]. Maszyna wirtualna jest uruchamiana na hypervisorze, czyli oprogramowaniu, które udaje sprzęt komputerowy. Takie rozwiązanie pozwala uruchomić zupełnie inny system operacyjny niż jest używany do obsługi serwerów chmury. Kontenery z kolei są pojemnikami na procesy, których zadaniem jest odizolowanie procesów w kontenerze zarówno od procesów systemu macierzystego jak i od procesów będących w innych kontenerach. Jak widać wspólną częścią kontenerów i systemu macierzystego jest jądro systemu operacyjnego. W przypadku maszyn wirtualnych system zainstalowany na tej maszynie posiada własne jądro i tylko korzysta z zasobów systemu macierzystego.

W niniejszym seminarium zamierzam przedstawić przykładową budowę takich chmur opartych na komunikacji klient-serwer z użyciem kontenerów OpenVZ [6-7]. Prowadzone przeze mnie badania uwzględniały scenariusz badający co widzi klient jak i co widzi serwer. Pod koniec wystąpienia przedstawię postęp w swoich obecnych pracach nad użyciem projektu CRIU na systemie Linux do migracji pojedynczych procesów korzystających z sieci komputerowej.

Literatura:

1. Hacker, Thomas J., Fabian Romero, and Jeremiah J. Nielsen. "Secure live migration of parallel applications using container-based virtual machines." *International Journal of Space-Based and Situated Computing* 2.1 (2012): 45-57.
2. Tafa, Igli, et al. "The evaluation of transfer time, cpu consumption and memory utilization in XEN-pv, XEN-hvm, openvz, kvm-fv and kvm-pv hypervisors using ftp and http approaches." *Intelligent Networking and Collaborative Systems (INCoS), 2011 Third International Conference on*. IEEE, 2011.
3. Clark, Christopher, et al. "Live migration of virtual machines." *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005.
4. Hines, Michael R., Umesh Deshpande, and Kartik Gopalan. "Post-copy live migration of virtual machines." *ACM SIGOPS operating systems review* 43.3 (2009): 14-26.
5. Soltesz, Stephen, et al. "Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors." *ACM SIGOPS Operating Systems Review*. Vol. 41. No. 3. ACM, 2007.
6. Mirkin, Andrey, Alexey Kuznetsov, and Kir Kolyshkin. "Containers checkpointing and live migration." *Proceedings of the Linux Symposium*. 2008.
7. Romero, Fabian, and Thomas J. Hacker. "Live migration of parallel applications with openvz." *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*. IEEE, 2011.