

Rozszerzanie systemów internetowych

Projektowanie i programowanie systemów internetowych I

mgr inż. Krzysztof Rewak

4 czerwca 2018

Wydział Nauk Technicznych i Ekonomicznych

Państwowa Wyższa Szkoła Zawodowa im. Witelona w Legnicy

Plan prezentacji

1. Skalowanie systemów
2. Automatyzacja wdrażania systemów
3. Podsumowanie

Skalowanie systemów

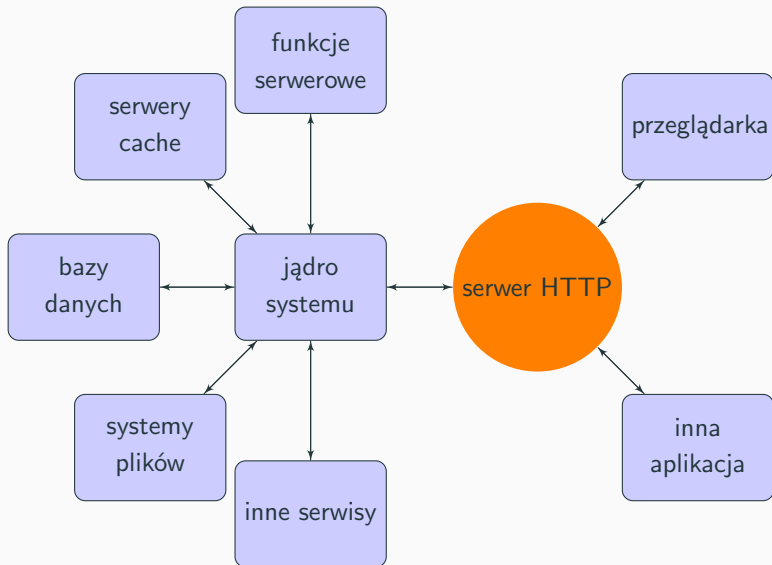
Czym jest **system**?

W teorii systemów jest to zespół elementów realizujący pewną określoną funkcję.

Czym jest **system**?

W rozumieniu systemów internetowych jest to zestaw serwisów tworzących aplikację.

Budowa klasycznego systemu internetowego



Czym jest **skalowanie**?

W matematyce możemy powiedzieć o działaniu mnożenia wektora V przez skalar K (jako funkcję $K \times V \rightarrow V$).

Czym jest **skalowanie**?

W systemach internetowych chodzi o możliwość zwiększenia skali działania systemu.

Jak skalować?

Skalowanie systemów IT można podzielić na dwa podstawowe typy:

- poziome, *scale-out*, w bok;
- pionowe, *scale-up*, w górę.

Skalowanie w górę

Skalowanie w górę jest rozwiązaniem doraźnym, relatywnie łatwym do zastosowania, ale w na dłuższą metę kosztowniejszym.

Skalowanie w górę

Skalowanie w górę polega na zwiększaniu zasobów pod względem ich parametrów.

Przykładowo jest to dokupienie i instalacja:

- mocniejszego procesora,
- mocniejszej karty graficznej,
- większego dysku twardego.

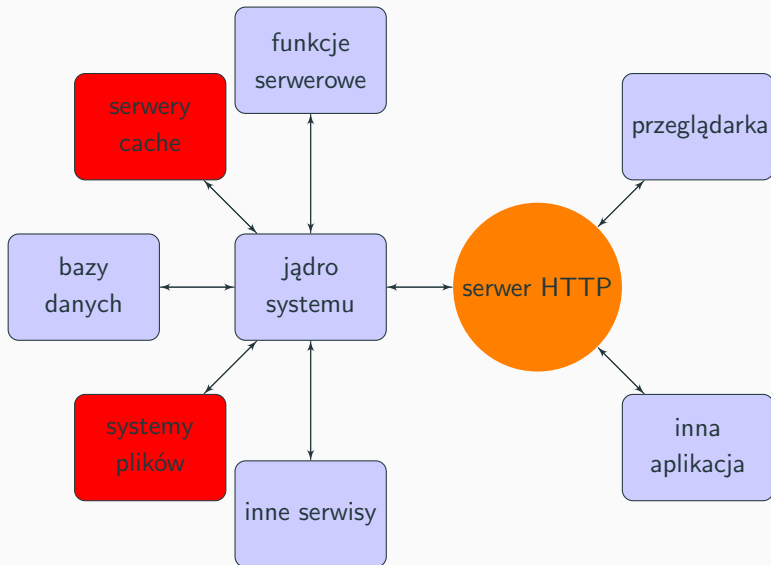
Skalowanie w górę

Skalowanie w górę może być łatwym rozwiązaniem problemów.

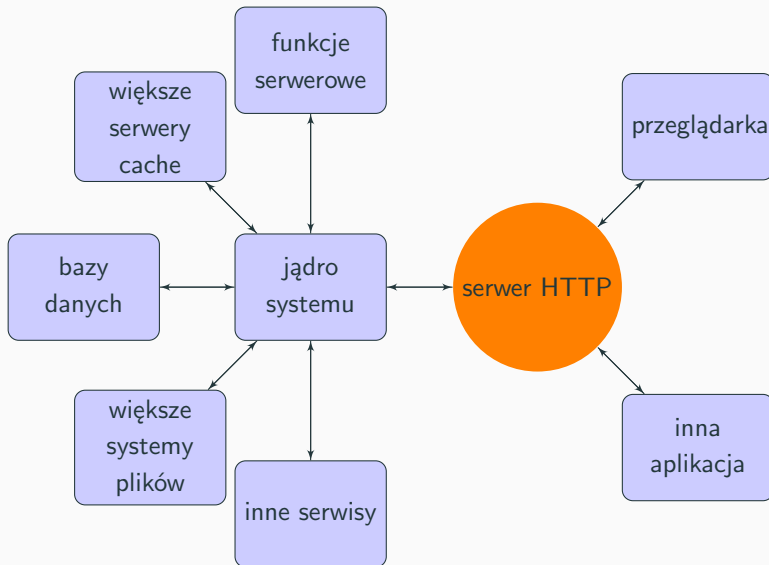
Przykładowo jest to dokupienie i instalacja:

- kończy się miejsce na dysku? zwiększmy limity!
- procesor nie wyrabia? podkręćmy go!

Budowa klasycznego systemu internetowego



Budowa klasycznego systemu internetowego



Skalowanie w górę

Niestety początkowe niskie koszty stosowania skalowania w górę przeradzają się z czasem w problemy z wydajnością, wzrost stałych kosztów oraz barierę technologiczną.

Skalowanie w bok

Skalowanie w bok jest rozwiązaniem długoterminowym, niekiedy nieco trudniejszym w implementacji, ale ostatecznie zazwyczaj zalecanym.

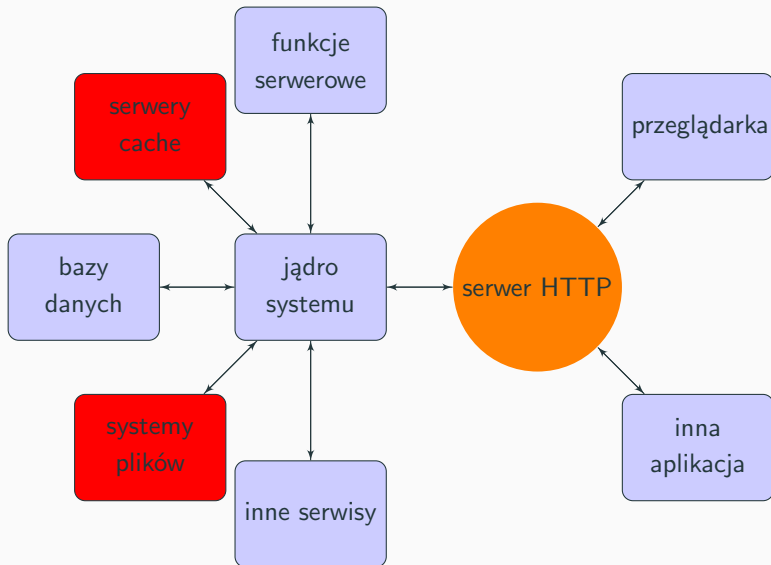
Skalowanie w bok

Skalowanie w bok polega na zwiększaniu zasobów pod względem ich liczby.

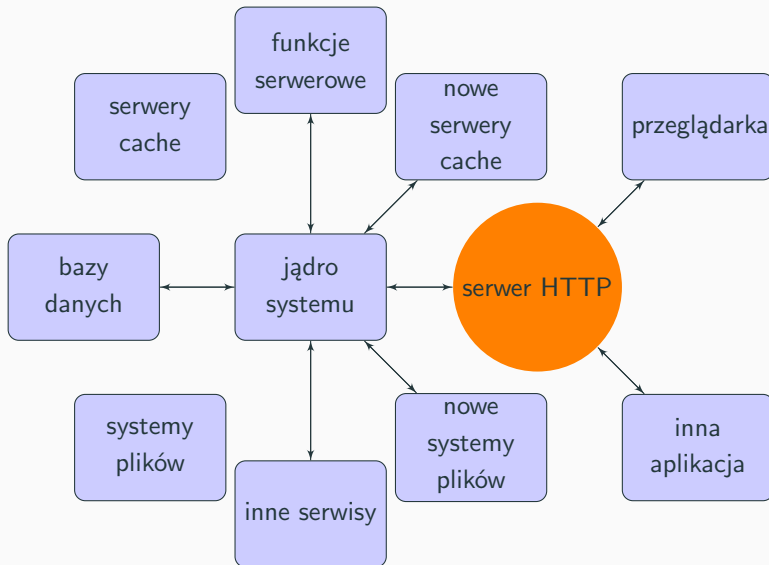
Przykładowo jest to dokupienie i instalacja:

- nowego procesora,
- nowej karty graficznej,
- nowego dysku twardego.

Budowa klasycznego systemu internetowego



Budowa klasycznego systemu internetowego



Skalowanie w bok

Skalowanie w bok jest łatwe i z czasem mniej kosztowne od skalowania w górę, jednakże jest trudniejsze w implementacji.

Przykładowo jak rozpraszać dane na kilka baz danych?

Automatyzacja wdrażania systemów

Jak wdrożyć system internetowy?

Przy wdrażaniu projektu można obrać wiele dróg. Między innymi:

- zbudować go w pełni na maszynie deweloperskiej i przesać przez FTP? (brzmi okropnie)
- wykorzystać system kontroli wersji, skorzystać z SSH i odpalić wszystko, co trzeba na maszynie produkcyjnej (nieco lepiej, ale też może boleć)
- oskryptować wszystko i uruchomić wdrożenie pod jednym przyciskiem.

Jak wdrożyć system internetowy?

Continuous integration, **CI**, czyli *ciągła integracja* to jedna z części *extreme programming*, zaraz obok testów jednostkowych, programowania parami czy stand-upów.

Dobre praktyki CI

Na *continuous integration* składa się kilka dobrych praktyk, o których warto wspomnieć.

Korzystanie z systemu kontroli wersji to podstawa. VCS powinien zarządzać repozytoriami z kodem aplikacji. Główna gałąź repozytorium powinna zawierać najaktualniejszą wersję aplikacji.

Jeżeli to potrzebne, należy zautomatyzować proces budowania aplikacji. Tutaj warto zapoznać się z oprogramowaniem takim jak Travis, Jenkins lub Gitlab.

Dobre praktyki CI

Kod powinien być otestowany, a środowisko testów uruchamiane automatycznie. Przed wdrożeniem nie może się okazać, że niektóre części aplikacji działają w sposób inny niż oczekiwany.

W teorii każda zmiana powinna trafiać na główną gałąź repozytorium codziennie. Każdy commit powinien również inicjować proces budowania aplikacji.

Proces integracji powinien być szybki.

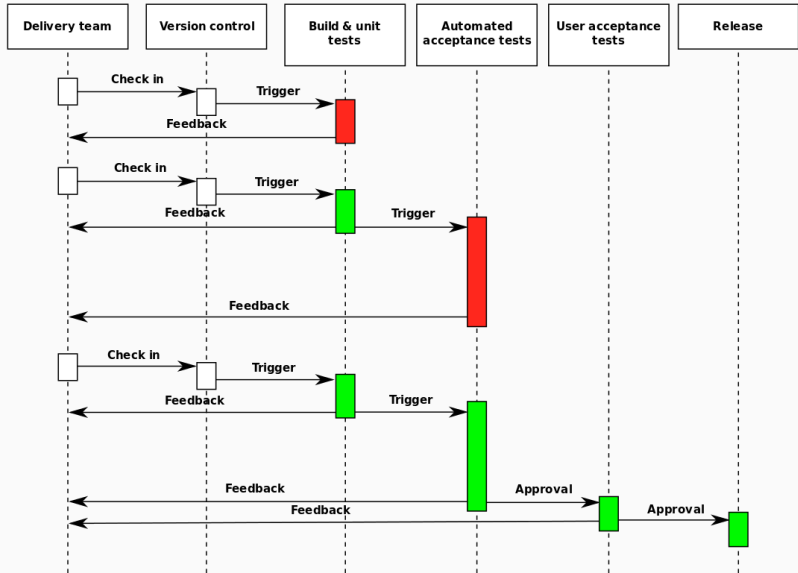
Testy powinny odbywać się w specjlanym środowisku. Może się ono nieznacznie różnić od produkcyjnego (przede wszystkim skalą), lecz idealnie byłoby wykorzystanie konteneryzacji lub wirtualizacji do maksymalnego zbliżenia warunków środowiskowych.

Dobre praktyki CI

Każdy powinien móc zobaczyć status ostatnich akcji. Mowa zarówno o testach automatycznych, pobraniu zależności, budowaniu aplikacji i innych funkcjonalnościach.

Wykorzystać *continuous deployment*, czyli pełne zautomatyzowanie procesu wdrażania aplikacji.

CD



Podsumowanie

Bibliografia i ciekawe źródła



https://en.wikipedia.org/wiki/Continuous_integration



https://en.wikipedia.org/wiki/Continuous_delivery

Pytania?

Kod prezentacji dostępny jest w repozytorium git pod adresem
<https://bitbucket.org/krewak/pwsz-ppsi>



Wszystkie informacje dot. kursu dostępne są pod adresem
<http://pwsz.rewak.pl/kursy/4>

