

naszaklasa
MIEJSCE SPOTKAŃ

Studium przypadku budowania
skalowalnych stron www

Tomasz Paszkowski

Agenda

- » Dedykowana serwerownia
- » Internet
- » Zakupy
- » Rezerwa awaryjna
- » Zarządzanie Infrastruktura
- » Proces Tworzenia Oprogramowania

Agenda

- » Architektura Oprogramowania
- » Planowanie pojemności
- » Architektura Systemów

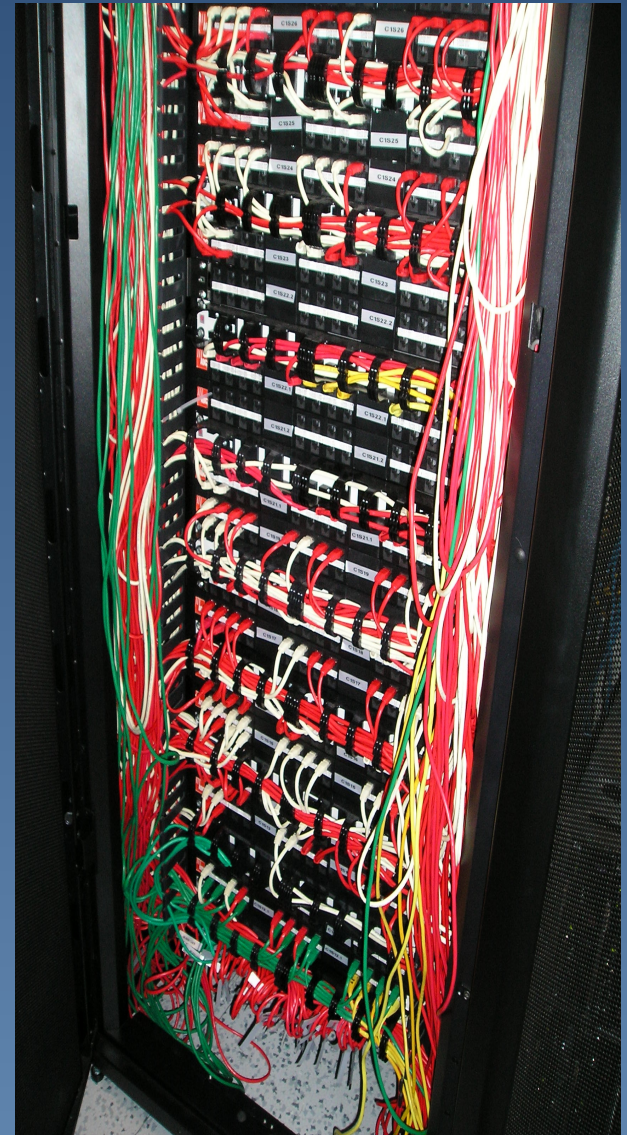
Dedykowana serwerownia

- » Neutralność we wszystkich aspektach (technologia, łączność, okablowanie)
- » Możliwość rozwoju i bezpłatnej rezerwacji powierzchni (kolokacja wielkopowierzchniowa)
- » Odpowiednia ilość dostępnej energii dostarczonej w bezpieczny sposób (UPS, Agregaty, dwie trafostacje)



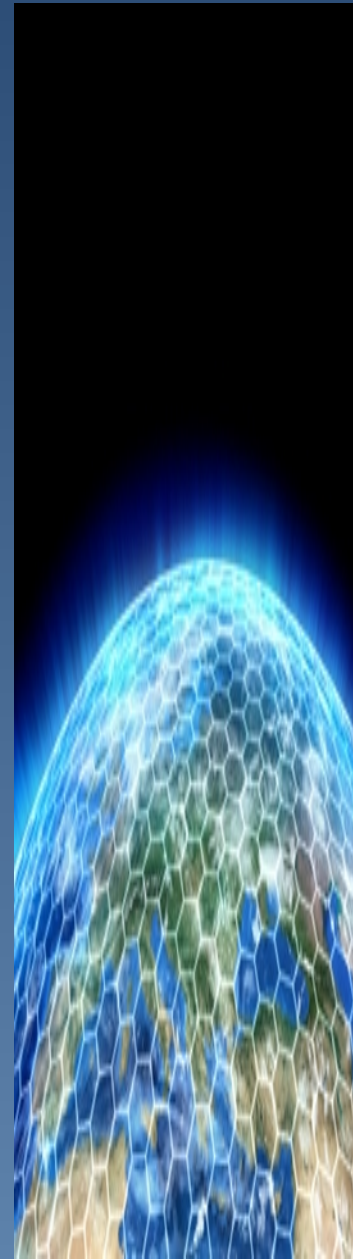
Dedykowana serwerownia

- » Klimatyzacja jej redundancja i wydajność
- » Gaszenie gazem, VESDA
- » Lokalizacja i rozładunek sprzętu i jego instalacja
- » Ceny (energia, klimatyzacja ,stanowiska), SLA
- » Przykłady w Polsce: Neutral Data Center (PLIX), Telehouse.Poland (ATM)



Internet

- » ASN
- » Oczywiście PLIX, ACX jako backup
- » TIER-1 (T-Systems, Level3, GBLX, Tata)
- » Telekomunikacja Polska (nadal)
- » Unikać pośredników (już od 1Gb/s), mała elastyczność kontraktów, wyższe ceny



Zakupy

- » Dwie strategie single oraz multi vendor
- » Standaryzacja wykorzystywanych komponentów
- » Kupować tylko w pełni wyposażone i obsadzone (całymi półkami, szafami) w dużych partiach
- » Wygrzewać (memtest, vdbench)
- » Pilnować gwarancji (min. 3 lata)
- » Zwróć uwagę na ceny serwisu po wygaśnięciu gwarancji



Zakupy

- » Przy zakupach realizowanych przy wsparciu vendora negocjować marże partnerskie
- » Jeżeli jest taka możliwość to kupować bezpośrednio z rynku dystrybucyjnego



Rezerwa awaryjna

- » Statystyczne podejście do awarii sprzętu
- » Stworzenie własnej rezerwy serwisowej
- » Odpowiednie planowanie eliminuje konieczność zakupu rozszerzonej gwarancji



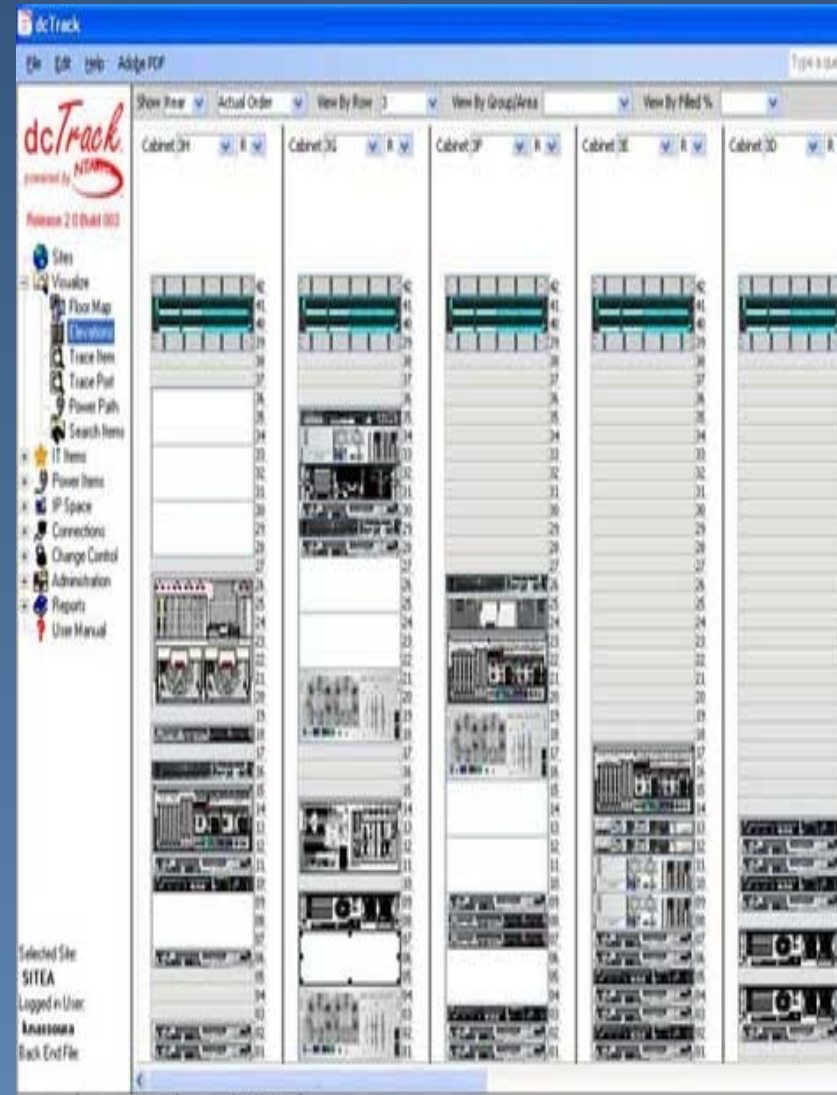
Zarządzanie infrastrukturą

- » Maksymalna zarządzalność każdego z elementów (PDU, Serwery, Storage, LAN)
- » Dedykowana sieć pod interfejsy MGMT
- » Backdoor firewall, VPN do sieci MGMT



Zarządzanie infrastrukturą

- » Inwentaryzacja wszystkiego (rozmieszczenie w szafach, wyposażenie serwerów macierzy, zajętość portów PDU, LAN, SAN czy wykorzystania zasobów LUN czy serwerów)
- » Na początek google spreadsheet, potem dcTrack, czy CCMDB (IBM Tivoli, HP UCMDB)
- » Narzędzia monitoringowe rozumiejące zależność pomiędzy usługami, systemami a infrastrukturą
- » Infrastruktura żyje bądź na to przygotowany



Proces Tworzenia Oprogramowania

- » Model iteracyjny
- » Zespoły projektowe
(Architekt, Administrator, Programista, Koder
CSS/HTML, Grafik, Tester)
- » PM odpowiedzialny za każdy aspekt projektu
- » Manager kontroli jakości prowadzi audyty projektów
- » Testy automatyczne (Selenium, JMeter, AppScan)
- » Code-review każdego fragmentu kodu
- » JIRA

Proces Tworzenia Oprogramowania

- » Indywidualne środowisko developerskie dla każdej osoby zaangażowanej w rozwój oprogramowania
- » Kilkanaście niezależnych środowisk testowych identycznych jak środowisko produkcyjne
- » Część nie rdzeniowych projektów prowadzonych przez firmy zewnętrzne
- » 3PP API (model zamknięty) na bazie OpenSocial

Architektura Oprogramowania

- » Awarie są i zawsze będą występować (ROC)
- » Projektowanie serwisów odpornych na awarie poszczególnych komponentów poprzez projektowanie ich jako niezależnych części
- » Zapobieganie propagowaniu błędów pomiędzy komponentami
- » Rozwiązywanie najczęściej występujących problemów
- » Kompatybilność wsteczna przydatna zwłaszcza przy rollbacku

Architektura Oprogramowania

- » Zapobieganie dublowaniu ten samej funkcjonalności w wielu komponentach
- » Unikanie stosowania skomplikowanych algorytmów
- » Partycjonowanie pionowe i poziome (dużo kubeków)
- » Unikanie stosowania rozwiązań SPOF
- » Wydajność to nie tylko serwery ale również przeglądarka

Architektura Oprogramowania

- » Wykorzystywanie trwałych pamięci podręcznych (REDIS, NKDB)
- » Dopasowanie technologii do rozwiązania (PHP, C, C++, Java, Erlang)

Architektura Oprogramowania

- » Metryki wydajności dla każdego z komponentów na poziomie aplikacji pozwalające obserwować w czasie rzeczywistym zachowanie aplikacji
- » Metryki czasów odpowiedzi poszczególnych komponentów systemu, operacji czy elementów zewnętrznych
- » Metryki błędów zaimplementowane w mechanizmach obsługi wyjątków

Planowanie pojemności

- » Zrozumienie zależności pomiędzy ilością korzystających użytkowników a obciążeniem systemu (metryki)
- » Zapas pojemności na poziomie 50% pozwala bezpiecznie rozbudowywać system
- » Analiza trendów istotnym element procesu
- » Nie zawsze opłaca się optymalizować czasami taniej jest kupić więcej sprzętu

Planowanie pojemności

- » Przy nowych projektach lepiej przeskalować niż borykać się z brakiem sprzętu do obsłużenia ruchu
- » Liczy się nie tylko CAPEX ale dla równie ważny jest OPEX (prąd, klimatyzacja, miejsce)
- » Warto brać pod uwagę zastąpienie starego sprzętu nowym (dual-core vs westmere six core)

Architektura Systemów

- » Redundancja tylko tam gdzie potrzebna
- » Szybkie przywracanie do pożądanego stanu po awarii
- » Automatyczne propagowanie zmian na serwery (Itsp,fai,puppet)
- » Zarządzanie rolami i profilami a nie serwerami (maksymalna automatyzacja)
- » Replikacja jako podstawowy element zapewniający bezpieczeństwo danych

Architektura systemów

- » Unikanie kaskadowania wielu serwisów na jednej maszynie
- » Wykorzystywanie tylko znanego i dobrze przetestowanego oprogramowania
- » Błędy poszczególnych komponentów nie propagują się do pozostałych
- » Zarządzanie zmianami
- » Proste rozwiązania ułatwiają diagnostykę