

Architektura oprogramowania i podejmowanie decyzji

Wykorzystywanie przywództwa, technologii
i zarządzania produktem do budowy
świetnych produktów

Srinath Perera

Przekład: Krzysztof Kapustka

APN Promise Warszawa 2024

Spis treści

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <i>Od autora</i> | <i>xi</i> |
| 1 Wprowadzenie do przywództwa w oprogramowaniu | 1 |
| Rola oceny sytuacji | 1 |
| Cel tej książki | 3 |
| Część I: Wprowadzenie | 8 |
| Część II: Podstawowe informacje | 8 |
| Część III: Projekt systemu | 9 |
| Część IV: Łączenie wszystkiego w całość | 10 |
| 2 Systemy, projekt i architektura | 11 |
| Czym jest architektura oprogramowania? | 11 |
| Jak zaprojektować system? | 13 |
| Pięć pytań | 15 |
| Pytanie 1: Kiedy jest najlepszy czas na wprowadzenie produktu na rynek? | 15 |
| Pytanie 2: Jaki jest poziom umiejętności zespołu? | 16 |
| Pytanie 3: Jaka jest wrażliwość naszego systemu na wydajność? | 17 |
| Pytanie 4: Kiedy możemy przepisać system? | 18 |
| Pytanie 5: Jakie są trudne problemy? | 19 |
| Siedem zasad: koncepcje nadrzędne | 19 |
| Zasada 1: Opieraj wszystko na doświadczeniu użytkownika | 19 |
| Zasada 2: Używaj iteracyjnej strategii cienkich plastrów | 21 |
| Zasada 3: W każdej iteracji dodawaj największą wartość przy jak najmniejszym wysiłku w celu wsparcia większej liczby użytkowników | 22 |
| Zasada 4: Podejmuj decyzje i absorbuj ryzyko | 25 |
| Zasada 5: Projektuj dogłębnie rzeczy, które trudno zmienić, ale implementuj je powoli | 25 |
| Zasada 6: Wyeliminuj niewiadome i wyciągaj wnioski z dowodów, pracując nad trudnymi problemami wcześniej i równoległe | 27 |
| Zasada 7: Poznaj kompromisy między spójnością i elastycznością w architekturze oprogramowania | 28 |
| Projektowanie księgarń internetowej | 30 |
| Projektowanie dla chmury | 34 |
| Podsumowanie | 36 |

| | |
|------------------------------------------------------------------------------------------------------------------|-----------|
| 3 Modele mentalne pozwalające zrozumieć i wyjaśnić wydajność systemu. . . | 37 |
| System komputerowy. | 39 |
| Modele dla wydajności. | 40 |
| Model 1: Koszt przełączenia z trybu użytkownika na tryb jądra | 40 |
| Model 2: Hierarchia operacji | 41 |
| Model 3: Narzut związany z przełączaniem kontekstu | 42 |
| Model 4: Prawo Amdahla | 42 |
| Model 5: Prawo skalowalności uniwersalnej | 43 |
| Model 6: Kompromisy dotyczące opóźnienia i stopnia wykorzystania | 44 |
| Model 7: Projektowanie pod kątem przepustowości z użyciem modelu maksymalnego użytecznego wykorzystania. | 44 |
| Model 8: Dodawanie limitów opóźnień. | 46 |
| Techniki optymalizacji | 49 |
| Techniki optymalizacji procesora | 50 |
| Techniki optymalizacji operacji I/O | 51 |
| Techniki optymalizacji pamięci. | 53 |
| Techniki optymalizacji opóźnień. | 54 |
| Intuicyjne wycucie wydajności | 55 |
| Uwarunkowania przywództwa | 56 |
| Podsumowanie | 57 |
| 4 Doświadczenia użytkownika (UX) | 59 |
| Ogólne koncepcje UX dla architektów | 59 |
| Zasada 1: Zrozum swoich użytkowników | 60 |
| Zasada 2: Rób możliwie jak najmniej | 61 |
| Zasada 3: Dobre produkty nie potrzebują instrukcji: ich użycie jest oczywiste. | 61 |
| Zasada 4: Myśl w kategoriach wymiany informacji | 62 |
| Zasada 5: Niech rzeczy proste będą proste. | 63 |
| Zasada 6: Zaprojektuj UX przed implementacją. | 63 |
| Projekt UX dla konfiguracji | 64 |
| Projekt UX dla API. | 66 |
| Projekt UX dla rozszerzeń | 68 |
| Rozważania dotyczące przywództwa | 69 |
| Podsumowanie | 70 |
| 5 Makroarchitektura: Wprowadzenie | 71 |
| Historia makroarchitektury. | 72 |
| Nowoczesne architektury | 75 |
| Elementy składowe makroarchitektury | 76 |
| Rozważania dotyczące przywództwa | 79 |
| Podsumowanie | 82 |

| | |
|--------------------------------------------------------------------------------------------|------------|
| 6 Makroarchitektura: koordynacja | 83 |
| Podejście 1: Rozpoczynaj przepływ z poziomu klienta | 83 |
| Podejście 2: Użyj innej usługi | 84 |
| Podejście 3: Użyj scentralizowanego oprogramowania pośredniczącego | 85 |
| Podejście 4: Zaimplementuj choreografię | 86 |
| Rozważania dotyczące przywództwa | 87 |
| Podsumowanie | 88 |
| 7 Makroarchitektura: zachowywanie spójności stanu | 89 |
| Dlaczego transakcje? | 89 |
| Dlaczego musimy wykroczyć poza transakcje? | 90 |
| Wykraczanie poza transakcje | 92 |
| Podejście 1: Ponowne zdefiniowanie problemu w celu obniżenia wymaganych gwarancji | 92 |
| Podejście 2: Stosowanie kompensacji | 93 |
| Najlepsze praktyki | 96 |
| Rozważania dotyczące przywództwa | 97 |
| Podsumowanie | 99 |
| 8 Makroarchitektura: obsługiwane bezpieczeństwo | 101 |
| Zarządzanie użytkownikami | 102 |
| Zabezpieczenia interakcji | 105 |
| Techniki uwierzytelniania | 107 |
| Techniki autoryzacji | 108 |
| Typowe scenariusze interakcji w zakresie zabezpieczeń dla aplikacji | 112 |
| Przechowywanie, RODO i inne regulacje | 115 |
| Strategia i porady dotyczące bezpieczeństwa | 118 |
| Wydajność i opóźnienie | 119 |
| Model zerowego zaufania | 120 |
| Zachowajmy ostrożność podczas uruchamiania kodu dostarczonego przez użytkownika | 121 |
| Tematy związane z blockchain | 121 |
| Inne tematy | 122 |
| Rozważania dotyczące przywództwa | 122 |
| Podsumowanie | 125 |
| 9 Makroarchitektura: obsługiwane wysokiej dostępności i skali | 127 |
| Dodawanie wysokiej dostępności | 127 |
| Replikacja | 127 |
| Szybkie odzyskiwanie | 130 |
| Skalowalność | 132 |
| Skalowanie dla nowoczesnej architektury: rozwiązanie podstawowe | 134 |
| Skalowanie: narzędzia pracy | 135 |

| | |
|------------------------------------------------------------------------------------|------------|
| Taktyka skalowania 1: Brak współdzielenia | 136 |
| Taktyka skalowania 2: Dystrybucja | 136 |
| Taktyka skalowania 3: Buforowanie | 137 |
| Taktyka skalowania 4: Przetwarzanie asynchroniczne | 137 |
| Budowanie skalowalnych systemów | 137 |
| Podejście 1: Sukcesywna eliminacja wąskich gardeł | 138 |
| Podejście 2: Projekt typu „shared-nothing” | 140 |
| Rozważania dotyczące przywództwa | 142 |
| Podsumowanie | 143 |
| 10 Makroarchitektura: rozważania na temat mikrousług | 145 |
| Decyzja 1: Obsługiwanie współdzielonych baz danych | 146 |
| Rozwiązanie 1: Jedna mikrousługa aktualizująca bazę danych | 147 |
| Rozwiązanie 2: Dwie mikrousługi aktualizujące bazę danych | 148 |
| Decyzja 2: Zabezpieczanie mikrousług | 148 |
| Decyzja 3: Koordynowanie mikrousług | 149 |
| Decyzja 4: Unikanie piekła zależności | 149 |
| Kompatybilność wsteczna | 149 |
| Kompatybilność w przód | 150 |
| Wykresy zależności | 151 |
| Luźno powiązane zespoły oparte na repozytoriach jako alternatywa dla mikrousług .. | 152 |
| Rozważania dotyczące przywództwa | 153 |
| Podsumowanie | 154 |
| 11 Architektury serwerów | 157 |
| Tworzenie usługi | 157 |
| Najlepsze praktyki w zakresie tworzenia usługi | 158 |
| Techniki zaawansowane | 160 |
| Korzystanie z alternatywnych modeli I/O i wątków | 160 |
| Narzut związany z koordynacją | 167 |
| Wydajne zapisywanie lokalnego stanu | 169 |
| Wybór systemu transportu | 170 |
| Obsługa opóźnień | 171 |
| Rozdzielanie odczytów i zapisów | 171 |
| Używanie blokad (i sygnalizacji) w aplikacjach | 172 |
| Korzystanie z kolejek i pul | 173 |
| Obsługa wywołań usług | 174 |
| Stosowanie powyższych technik w praktyce | 174 |
| Aplikacje ograniczone procesorem (procesor >> pamięć i brak operacji I/O) | 175 |
| Aplikacje ograniczone pamięcią (procesor + pamięć i brak operacji I/O) | 175 |
| Aplikacje zrównoważone (procesor + pamięć + operacje I/O) | 176 |
| Aplikacje ograniczone operacjami I/O (operacje I/O + pamięć > procesor) | 177 |
| Inne kategoryzacje aplikacji | 177 |

| | |
|-------------------------------------------------------------------------|------------|
| Rozważania dotyczące przywództwa | 178 |
| Podsumowanie | 180 |
| 12 Budowanie stabilnych systemów | 181 |
| Dlaczego systemy ulegają awariom i co możemy z nimi zrobić? | 181 |
| Jak obsługiwać znane błędy? | 183 |
| Nieoczekiwane obciążenie | 183 |
| Awarie zasobów | 188 |
| Zależności | 192 |
| Zmiany dokonywane przez ludzi | 193 |
| Typowe błędy | 194 |
| Wycieki zasobów | 194 |
| Zakleszczenia i powolne operacje | 195 |
| Jak obsługiwać nieznanne błędy? | 196 |
| Obserwowalność | 196 |
| Błędy i testowanie | 197 |
| Łaskawa degradacja | 199 |
| Rozważania dotyczące przywództwa | 199 |
| Podsumowanie | 201 |
| 13 Budowanie i ewoluowanie systemów | 203 |
| Rozpoczynanie pracy | 203 |
| Dbanie o podstawy | 203 |
| Zrozumienie procesu projektowania | 206 |
| Podejmowanie decyzji i absorbowanie ryzyka | 209 |
| Wymaganie doskonałości | 210 |
| Komunikowanie projektu | 212 |
| Ewolucja systemu: jak uczyć się od użytkowników i ulepszać system | 213 |
| Rozważania dotyczące przywództwa | 217 |
| Podsumowanie | 218 |
| <i>Indeks</i> | 221 |