

Dr inż. Tomasz Walkowiak

Instytut Informatyki, Automatyki i Robotyki
Politechnika Wrocławska
Wybrzeże Wyspiańskiego 27, 50-370, Wrocław
E-mail: Tomasz.Walkowiak@pwr.wroc.pl

SYMULACYJNA OCENA GOTOWOŚCI USŁUG SYSTEMÓW INTERNETOWYCH Z REALISTYCZNYM MODELEM CZASU ODNOWY

Słowa kluczowe: *usługa internetowa, gotowość, niezawodność, symulacja Monte-Carlo*

Abstrakt: W artykule przedstawiono metodę numerycznego wyznaczania zmian gotowości usług internetowych w czasie. Bierze ona pod uwagę aspekty niezawodnościowe i funkcjonalne systemu komputerowego świadczącego usługi. Analiza niezawodnościowa pozwala na wyznaczenie prawdopodobieństwa, że system będzie zdalny w danym momencie. Uwzględnia ona strukturę systemu komputerowego, losowe czasy do uszkodzeń (sprzętowych jak i oprogramowania związanych z naruszeniami zabezpieczeń) oraz szczegółowy model odnowy biorący pod uwagę godziny pracy administratorów oraz czas związany z dostarczeniem elementów do wymiany. Analiza funkcjonalna pozwala na wyznaczanie prawdopodobieństwa, że użytkownik zostanie obsłużony w czasie mniejszym niż zadany. Oparta jest ona na modelowaniu procesu realizacji usługi jako sekwencji zadań wykonywanych na jednym lub kilku komputerach. Bierze ona pod uwagę zmienność intensywności napływu użytkowników w ciągu tygodnia oraz parametry takie jak: sekwencję zadań, alokację zadań na komputerach oraz parametry techniczne komputerów i zadań. Opisana metoda była podstawą do stworzenia aplikacji komputerowej wyznaczającej techniką symulacji Monte-Carlo zmienność gotowości systemu w ciągu tygodnia. Artykuł zawiera numeryczne rezultaty przykładowej analizy.

1. Wprowadzenie

Usługi świadczone przez Internet są w obecnym czasie podstawą funkcjonowania bardzo wielu przedsiębiorstw. Ich poprawne i nieprzerwane działanie jest istotnym aspektem sukcesu biznesowego. Użytkownicy serwisów internetowych mogą w bardzo prosty sposób przenieść się na inny portal. Dlatego, też istotne są metody pozwalające na ocenę gotowości systemu internetowego, a przede wszystkim metody pozwalające na zbadanie wpływu różnych parametrów związanych z konfiguracją i sposobem utrzymania systemu na jego gotowość. Gotowość w przypadku usług internetowych rozumiana jest najczęściej jako prawdopodobieństwo, że system jest zdalny w danym momencie czasu. Do zdefiniowania zdalności usługi internetowej wyjdziemy z definicji poprawnego działania systemu z pracy [29]: „system działa poprawnie wtedy, gdy wymagane zadania będą wykonywane poprawnie w założonym czasie”. Zawarte są niej dwa istotne aspekty analizowane w niniejszym artykule. Po pierwsze aspekt niezawodnościowy, system musi działać. Nie może być on uszkodzony. Po drugie, aspekt funkcjonalny. Fakt, że czas odpowiedzi systemu internetowego musi się mieścić w pewnych limitach czasowych. Użytkownik, który będzie czekał na reakcję usługi internetowej dłużej niż kilka sekund będzie traktował ją jako nie działającą i jeżeli będzie taka możliwość poszuka innej usługi w Internecie o podobnej funkcjonalności.

Istnieje szereg prac analizujących niezawodność usług internetowych. Jednak w większości model niezawodnościowy systemu oparty jest na procesach Markowa [3] i system

usług sprowadzany jest do prostego systemu szeregowego [18]. W niniejszej pracy proponujemy skupić się na dokładnym zamodelowaniu sposobu przeprowadzania napraw w systemie internetowym biorąc pod uwagę między innymi godziny pracy pracowników i wolne weekendy. Analiza aspektu funkcjonalnego oparta jest o często stosowane rozwiązanie w analizie systemów technicznych o dużej złożoności: modelowanie i symulację [4]. Zaproponowany został model realizacji usług internetowych oparty na obserwacji zadań wykonywanych na poszczególnych komputerach [26]. Model ten jest podstawą do budowy symulatora, który wykorzystując technikę Monte-Carlo [8] pozwala na wyznaczenie prawdopodobieństwa, że użytkownik systemu zostanie obsłużony w zadanym czasie przy zadanym obciążeniu usługi internetowej. Modelując intensywność napływu użytkowników, która jest w praktyce zmienna w czasie i biorąc pod uwagę aspekty niezawodnościowe można wyznaczyć gotowość usługi internetowej.

2. Stan wiedzy i pokrewne prace

Analiza niezawodności, w tym gotowości złożonych systemów (do których należą systemy internetowe) są tematyką badań od wielu lat. Tematyka ta opisywana jest w wielu bardzo dobrych podręcznikach, przykładowo w popularnej w świecie anglosaskim książce Barlowa i Proshana [3]. Problem modelowania czasu odnowy jest najczęściej rozwiązywany poprzez użycie rozkładu wykładniczego [1, 3, 18]. Wynika to przede wszystkim z użyciu procesów Markowa w analizie niezawodnościowej systemu. Stosunkowo rzadziej analizowane są przypadki z innymi rozkładami czasu naprawy. Przykładowo w pracach [13, 14] zakłada się, że czas naprawy jest opisywany procesem geometrycznym (zaproponowanym przez Lama w pracy [16]). Najczęściej zakłada się [1, 3], że czas naprawy rozpoczyna się natychmiast po wystąpieniu uszkodzenia. Aczkolwiek, analizowane są w literaturze również inne scenariusze [7, 30] zakładające, że od momentu wystąpienia uszkodzenia do zakończenia naprawy występują dwa przedziały czasu: czas oczekiwania oraz rzeczywisty czas naprawy. Czas oczekiwania modelowany jest jako zmienna losowa niezależna od czasu życia komponentu. Analizuje się również przypadki mieszane [14], w których czas naprawy z pewnym prawdopodobieństwem składa się z czasu oczekiwania i rzeczywistego czasu naprawy lub tylko z rzeczywistego czasu naprawy.

Podójście analityczne (wspominane powyżej) posiada istotne ograniczenia [21] co do użytych rozkładów (przede wszystkim wykładniczych), ograniczonego zastosowania (tylko do wybranej grupy systemów) i słabej zgodności z rzeczywistością. Dlatego też stosuje się podejście symulacyjne [4]. Wykorzystanie symulacji komputerowej [5], a w szczególności symulacji Monte-Carlo [8] do analizy niezawodnościowej złożonych systemów posiada bogatą literaturę czego przykładem może być monografia [15] czy prace [6, 21].

Opisywane w literaturze metody wyznaczenie gotowości systemów internetowych bazują na wykorzystaniu modeli analitycznych [9, 12] lub monitorowaniu rzeczywistego systemu [2, 11, 22]. Natomiast opisana w niniejszej pracy metoda wyznaczenia gotowości bazuje na symulacyjnym wyznaczaniu czasu dostępu do serwisu internetowego w funkcji liczby obsługiwanych użytkowników (rozdział 5).

Czas dostępu do serwisów internetowych można wyznaczyć przeprowadzając testy wydajnościowe rzeczywistych systemów (przykładowe narzędzia to: Funkload, Apache JMeter, Rational Performance Tester czy Developer Tools od firmy Microsoft), budując modele analitycznego [19, 25] (najczęściej wykorzystując sieci kolejek [17]) lub metodą symulacyjną [24] (również często bazującą na modelach kolejkowych).

3. Gotowość usługi

Miarą jakości serwisów internetowych jest gotowość, rozumiana jako prawdopodobieństwo, że użytkownik pojawiający się w danej chwili zostanie obsłużony. Oznaczając takie zdarzenie losowe przez U_t można gotowość usługi formalnie zapisać jako:

$$A(t) = \Pr(U_t). \quad (1)$$

Wprowadzając kolejne oznaczenia: S_t - zdarzenie losowe, że system komputerowy, na którym uruchomiona jest usługa jest zdalny i korzystając z zależności na prawdopodobieństwa warunkowe wzór (1) można sprowadzić do postaci:

$$A(t) = \Pr(U_t | S_t) \Pr(S_t) + \Pr(U_t | \bar{S}_t) \Pr(\bar{S}_t). \quad (2)$$

W przypadku analizowanego typu systemów użytkownik serwisu nie zostanie obsłużony w momencie, gdy system komputerowy jest w stanie niezdatności (jest uszkodzony). Zatem $\Pr(U_t | \bar{S}_t) = 0$. Upraszcza nam to wzór na gotowość (2) do postaci:

$$A(t) = \Pr(U_t | S_t) \cdot \Pr(S_t). \quad (3)$$

Będzie on podstawą do opracowania metody obliczeń komputerowych opartych o symulację pozwalających na wyznaczanie wartości gotowości usługi. Dla zwiększenia czytelności dalszego wywodu wprowadźmy nazwy na poszczególne czynniki iloczynu (3). Niech $\Pr(S_t)$ będzie nazywane gotowością systemu, rozumianą, jako prawdopodobieństwo że system komputerowy w danej chwili jest zdalny, nie jest uszkodzony. Natomiast, $\Pr(U_t | S_t)$ nazwijmy gotowością funkcjonalną, rozumianą jako prawdopodobieństwo, że użytkownik pojawiający się w danej chwili zostanie obsłużony w założonym limicie czasowym (t_{max}) przy założeniu, że system komputerowy jest zdalny. Można, zatem gotowość funkcjonalną zapisać jako:

$$\Pr(U_t | S_t) = \Pr(\text{czasodpowiedzi}(T) < t_{max} | T = t \wedge S_t) . \quad (4)$$

4. Model niezawodnościowy systemu komputerowego

System komputerowy realizujący usługę internetową z punktu widzenia niezawodności można rozumieć jako system odnawialny. Dla uproszenia, założymy, że uszkodzenie, któregośkolwiek komponentu systemu (sprzętu czy też oprogramowania), będzie skutkowało przejściem całego systemu do stanu niezdatności [18].

Należy tutaj zaznaczyć, że najczęstszą przyczyną uszkodzeń współczesnych systemów internetowych są incydenty, tzn. naruszenia zabezpieczeń systemu w wyniku różnego typu ataków sieciowych. Incydenty obejmują zainfekowanie systemu przez oprogramowanie złośliwe: wirusy, robaki internetowe, itp. Tego typu incydenty określa się jako ataki nieukierunkowane, ponieważ zwykle nie są one skierowane przeciwko konkretnemu systemowi, węzłowi lub usłudze. Innym typem incydentów są próby włamania się do systemu z sieci. Bazują one na wykorzystaniu luk w zabezpieczeniach oprogramowania (usług, oprogramowania narzędziowego, systemu operacyjnego), a więc specyficznych usterek programistycznych. Oba typy uszkodzeń powodują, że system przestaje działać wiarygodnie, ponieważ nie ma pewności jakie fragmenty oprogramowania zostały zmodyfikowane.

Incydenty obejmują również ataki mające na celu doprowadzenie do upadku wybranej usługi, serwera lub węzła obliczeniowego, a w konsekwencji całego systemu. Są to różnego rodzaju techniki, mające na celu wyczerpanie zasobów węzła lub usługi. W efekcie usługa staje się niedostępna. Szczególnie znane są ataki DDOS (z ang. distributed denial-of-service), w których generuje się sztucznie zapotrzebowanie na usługę albo po prostu bezcelowy ruch

sieciowy, przychodzący do systemu od bardzo wielu adresów w sieci w tym samym czasie. W efekcie komputer staje się niedostępny dla legalnych użytkowników, a często również oprogramowanie upada z powodu wyczerpania zasobów.

Złożymy ponadto, że uszkodzenia są zdarzeniami niezależnymi i czas do uszkodzenia można zamodelować rozkładem wykładniczym. Natomiast proces odnowy zostanie zamodelowany dużo dokładniej biorąc pod uwagę czas pracy administratora systemu komputerowego w ciągu tygodnia oraz czas potrzebny na wymianę uszkodzonego sprzętu czy reakcję na zaistniały incydent. Założymy, że administrator systemu pracuje pięć dni w tygodniu w zadanych godzinach. I tylko w takich godzinach może nastąpić wykrycie awarii systemu i czynności związane z jego naprawą (np. reinstalacja oprogramowania). Zostanie również wzięty pod uwagę czas dostarczenia nowego sprzętu przez kuriera.

Podzielmy uszkodzenia na dwa rodzaje: wymagające dostawy nowego sprzętu oraz niewymagające tego (naprawy związane z oprogramowaniem czy też wymianą sprzętu na sprzęt dostępny na miejscu). Założymy, że z pewnym zadaniem prawdopodobieństwem pf występuje uszkodzenie pierwszego rodzaju (wymagające dostarczenia nowego sprzętu).

W procesie odnowy wyróżniono następujące elementy składowe (czas pracy administratora w przykładach obejmuje godziny od 8:00 do 17:00 z 1 godzinną przerwą):

- czas do wykrycia uszkodzenia – równy zero, gdy uszkodzenie miało miejsce w godzinach pracy administratora systemu, w przeciwnym przypadku równy czasowi do następnego dnia roboczego (przykładowo, gdy uszkodzenie nastąpiło o godzinie 17:00 w piątek, wynosi on 63 godziny, czyli czas do poniedziałku ósmej rano);
- czas analizy problemu – wyznaczany przez zmienną losową o obciążym rozkładzie wykładniczym, przy czym jeżeli moment zakończenia tej czynności wystąpiłby poza godzinami pracy, to jest on wydłużany o czas, w którym administrator nie pracuje, przykładowo jeżeli czynność analizy problemu miałaby trwać 4h, a rozpocznie się w poniedziałek o godzinie 14:00 to jej rzeczywisty czas trwania będzie wynosił 19h (do wtorku 9:00), czyli zostanie zwiększony o 15h przerwę w pracy;
- czas na dostarczenie elementu – z prawdopodobieństwem równym $1-pf$ równy zero (uszkodzenie niewymagające zamówienia elementów), w przeciwnym przypadku jest on określany jako czas do godziny 8:00 rano następnego dnia roboczego (czas na dostarczenie elementu przez kuriera);
- czas na naprawę – podobnie jak czas analizy problemu wyznaczany na podstawie wartości zmiennej losową o obciążym rozkładzie normalnym, powiększanej o czas kiedy administrator nie pracuje.

Przedstawiony model odnowy systemu komputerowego zawiera zmienne losowe o rozkładach wykładniczych, normalnych i dyskretnych. Jest on trudny do rozwiązania metodami analitycznymi (np. modelami Markowa). Dlatego, też autor zastosował metodę symulacji Monte-Carlo [8]. W celu automatyzacji obliczeń stworzono oprogramowanie symulacyjne w środowisku Prime SSF [20] wykorzystywanym do analizy czasu odpowiedzi usług internetowych (opisanej w następnym rozdziale). Działanie symulatora polega na wielokrotnym powtarzaniu symulacji procesu uszkodzeń i odnowy. Przy czym kolejne symulacje różnią się wartościami zmiennych losowych występujących w analizowanym systemie. Znając rozkład tych zmiennych (dane wejściowe) można uzyskać informację o zachowaniu systemu w większości możliwych (prawdopodobnych) sytuacjach. Poprzez obserwację zmian wartości interesujących użytkownika wielkości można uzyskać informację o ich rozkładzie i dowolnych statystykach. Pojedyncza symulacja polega na analizie funkcjonowania systemu w pewnym przedziale czasu (wielokrotność 7 dni), od punktu początkowego przez ściśle określoną długość.

Przedstawiony model niezawodnościowy zawiera pewne arbitralne założenia, np. wolny weekend czy też to, że jest jeden administrator i pracuje na jedną zmianę. Zastosowana metoda rozwiązania – symulacja komputerowa – pozwala na stosunkowo łatwą zmianę tych założeń na inne, ale wymaga to zmiany w kodzie źródłowym symulatora. Przykładowe wyniki liczbowe przedstawione są w rozdziale 6.

5. Model funkcjonalny usługi

5.1. Wyznaczanie czasu dostępu do serwisu internetowego

Jak to zostało przedstawione w rozdziale drugim, gotowość usług internetowych zależy od niezawodności systemu internetowego (omówionego w poprzednim rozdziale) i od tego jak szybko użytkownik będzie obsłużony przez system komputerowy. W celu wyznaczenia czasu dostępu do serwisu internetowego, przeanalizujemy proces wykonania żądania użytkownika przez system komputerowy realizujący wybraną usługę. Analiza będzie przeprowadzana z punktu widzenia realizacji usługi biznesowej realizowanej przez serwis [10]. Proces realizacji usługi można rozumieć jako sekwencję zadań wykonywanych na jednym lub kilku komputerach (węzłach). Zakładamy, że zadania wykonują się szeregowo, tzn. nowe zadanie w sekwencji obsługi danego żądania jest zakończone wtedy, gdy wszystkie zadania, które to żądanie wywołało zakończyły się. Sekwencja wywołań kolejnych zadań nazywana jest choreografią usługi (z ang. service choreography). Choreografię usługi u można przedstawić jako ciąg:

$$\text{choreografia}(u) = \left(c(\text{zadanie}_{b_1}), c(\text{zadanie}_{b_2}), \dots, c(\text{zadanie}_{b_n}) \right), \quad (5)$$

gdzie $c(\text{zadanie}_{b_i})$ reprezentuje zapytanie (\Rightarrow) skierowane do zadania zadanie_{b_i} lub odpowiedź (\Leftarrow) uzyskana po wykonaniu danego zadania. Część zadań po wykonaniu obliczeń zwraca odpowiedź do nadawcy. Inne zadania przed wysłaniem odpowiedzi mogą wywołać kolejne zadania. Każde zadanie umieszczone jest na którymś z komputerów systemu realizującego usługę.

Przyjmując powyższy model (szczegółowy opis modelu można znaleźć w pracy [26]) czas wykonania żądania użytkownika jest równy sumie czasu koniecznego do komunikacji pomiędzy komputerami, na których umieszczone są poszczególne zadania i czasu koniecznego na wykonanie zadań. Dla przykładowej choreografii:

$$\begin{aligned} \text{choreografia}(u) = \\ u \Rightarrow \text{zadanie}_1 \Rightarrow \text{zadanie}_2 \Leftarrow \text{zadanie}_1 \Rightarrow \text{zadanie}_3 \Leftarrow \text{zadanie}_1 \Leftarrow u \end{aligned} \quad (6)$$

i konkretnego rozmieszczenia zadań na komputerach, czas obsługi zapytania można przedstawić jako sumę:

$$\begin{aligned} \text{czasodpowiedzi}(u) = \text{czas}(h_0, h_1) + pt(\text{zadanie}_1) + \text{czas}(h_1, h_2) + pt(\text{zadanie}_2) + \\ \text{czas}(h_2, h_1) + \text{czas}(h_1, h_3) + pt(\text{zadanie}_3) + \text{czas}(h_3, h_1) + \text{czas}(h_1, h_0) \end{aligned}, \quad (7)$$

gdzie $\text{czas}(h_i, h_j)$ jest czasem transmisji zapytania lub odpowiedzi z komputera h_i do h_j , a $pt(\text{zadanie})$ jest czasem wykonania zadania. Dla współczesnych systemów internetowych niezwiązanych z transmisją audio- wideo można czas transmisji zapytań i odpowiedzi zamodelować przez zmienną losową o obciążym rozkładzie normalnym [26].

Czas wykonywania poszczególnych zadań zależy od rodzaju zadania (jego złożoności obliczeniowej), rodzaju komputera (jego parametrów technicznych) i od jego obciążenia – liczby i rodzaju innych zadań wykonywanych w tym samym czasie (np. w celu obsłużenia

zapytania od innego użytkownika). Ostatni z tych parametrów jest wartością zmienną w czasie i zależy od intensywności pojawiania się użytkowników. Jest go trudno wyznaczyć w sposób analityczny, dlatego też zastosowano technikę symulacji komputerowej opartej o technikę Monte-Carlo [8]. Symulator [27] został stworzony podobnie jak symulator omówiony w poprzednim rozdziale w środowisku Prime SSF [20]. Pozwala on dla zadanej choreografii (np. opisanej w języku WS-CDL), alokacji zadań na węzłach, konfiguracji węzłów (parametrów opisujących moc procesora i liczbę rdzeni), parametrów serwisów technicznych (rodzaju i konfiguracji serwerów WWW), złożoności obliczeniowej każdego z zadań oraz intensywności napływu klientów wyznaczyć czas wykonania każdego z symulowanych zadań.

5.2. Gotowość funkcjonalna

Wspomniany symulator pozwala na wyznaczenie w sposób numeryczny zależności prawdopodobieństwa, że użytkownik zostanie obsłużony w czasie krótszym niż zadany w funkcji intensywności napływu użytkowników:

$$Af(n) = \Pr(\text{czasodpowiedzi} < t_{\max} \mid \text{obciążenie} = n). \quad (8)$$

Wobec tego znając zmienność obciążenia usługi w czasie $\text{obciążenie}(t)$ można wyznaczyć gotowość funkcjonalną jako:

$$\Pr(U_t \mid S_t) = Af(\text{obciążenie}(t)). \quad (9)$$

Wartości funkcji zmienności obciążenia w czasie zależą od rodzaju oferowanej usługi. Przykładowo, serwisy internetowe przeznaczone dla studentów mają największy napływ użytkowników w okresie przed sesją i najczęściej w godzinach nocnych. W przykładowej analizie przedstawionej w kolejnym rozdziale, założono, że funkcja $\text{obciążenie}(t)$ jest okresowa, z okresem równym tygodniowi. Zmienność dobową i tygodniową jest często obserwowana w ogólnodostępnych serwisach. Można ją zauważyć analizując zmienność ruchu w sieciach dużych operatorów internetowych.

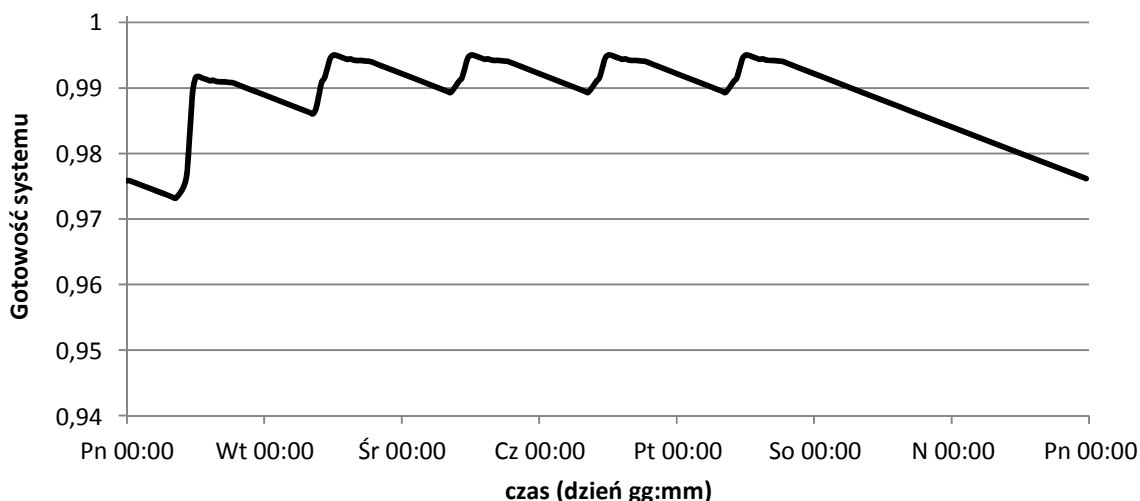
6. Analiza gotowości dla przykładowego systemu

6.1. Gotowość systemu

Przykładowy system poddany analizie składał się z sześciu węzłów, pięciu na których umieszczone były komponenty biznesowe opisane pewną choreografią i jednego rutera ze ścianą ogniową. Dzisiejsze komputery charakteryzują się wysoką niezawodnością dlatego też intensywność uszkodzeń (w 80% przypadków traktowanych jako wystąpienie incydentu, a w 20% uszkodzenia sprzętowego) ustawiono jako jedno uszkodzenie na rok. Parametry odnowy ustawiono następująco:

- średni czas analizy – 3 godziny;
- średni czas naprawy – 1 godzina;
- prawdopodobieństwo uszkodzenia sprzętowego (pf) – 0.2;
- czas pracy administratora od 8:00 do 17:00 z 1 godziną przerwą (12:00-13:00);
- odchylenie standardowe dla obciążonych rozkładów normalnych – 20% wartości średniej.

Symulację przeprowadzono dla 1.000 tygodni, powtarzając ją 50.000 razy. Wyniki przedstawione zostały na Rys. 1. Widać wyraźnie zmienność współczynnika gotowości systemu w ciągu tygodnia. Zauważyć można, że najniższy poziom gotowości jest w poniedziałek.



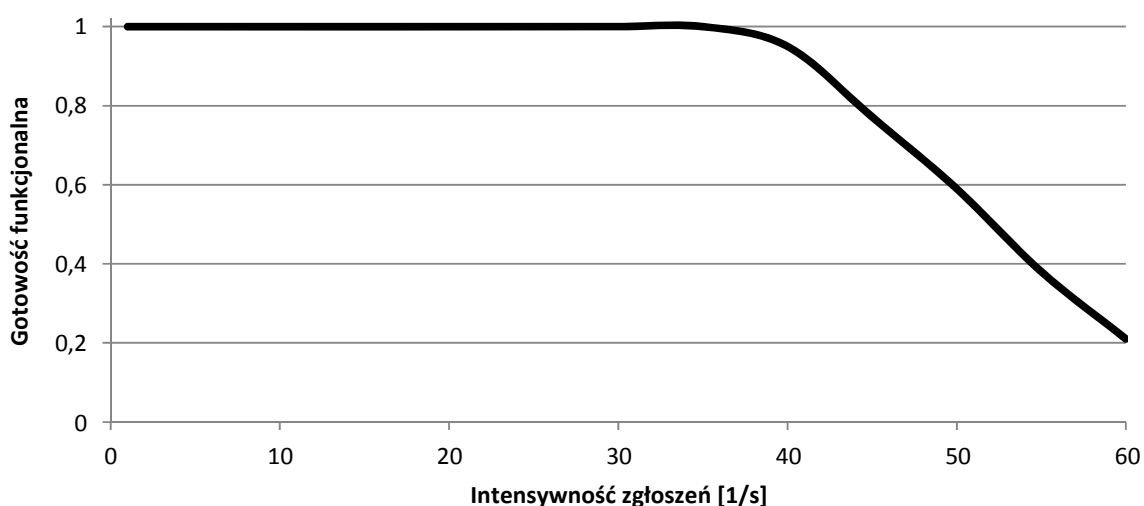
Rys. 1. Gotowość systemu w ciągu tygodnia dla przykładowego systemu

Zgadza się to z doświadczeniem autora jako administratora systemu internetowego. Spowodowane, jest to faktem, że w ciągu weekendu administratorzy nie pracują.

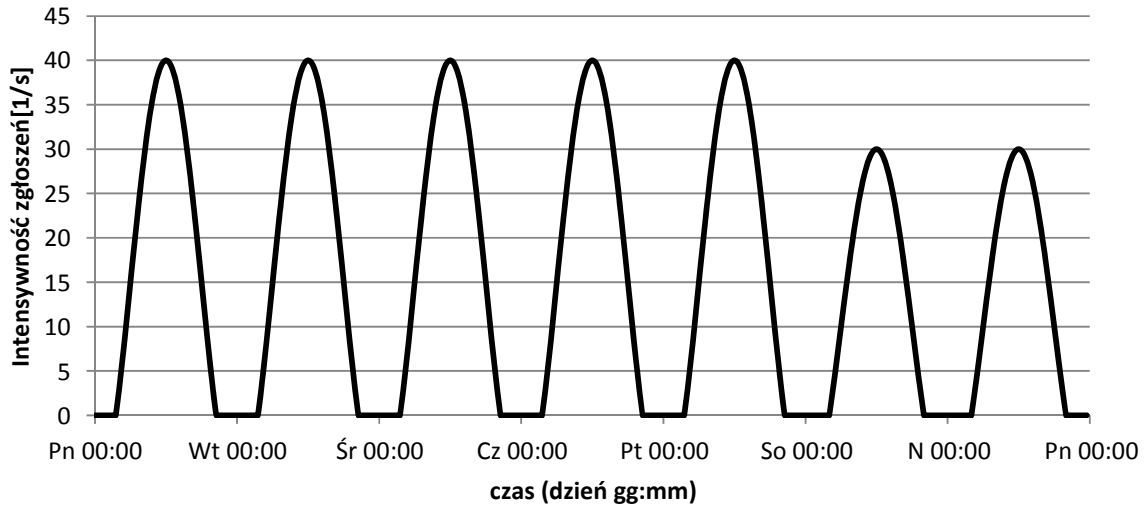
6.2. Gotowość funkcjonalna

Drugim elementem wpływającym na gotowość usługi jest gotowość funkcjonalna, rozumiana jako prawdopodobieństwo, że użytkownik usługi internetowej uzyska odpowiedź w czasie krótszym niż zadany limit przy założeniu, że system komputerowy realizujący usługę nie jest uszkodzony. Jak już wspomniano analizowany system składał się z 6 węzłów i realizował przykładową choreografię. Limit czasu obsługi ustawiony był na 12 sekund. Otrzymane wyniki symulacyjne, gotowości funkcjonalnej w funkcji intensywności użytkowników przedstawiono na Rys. 2.

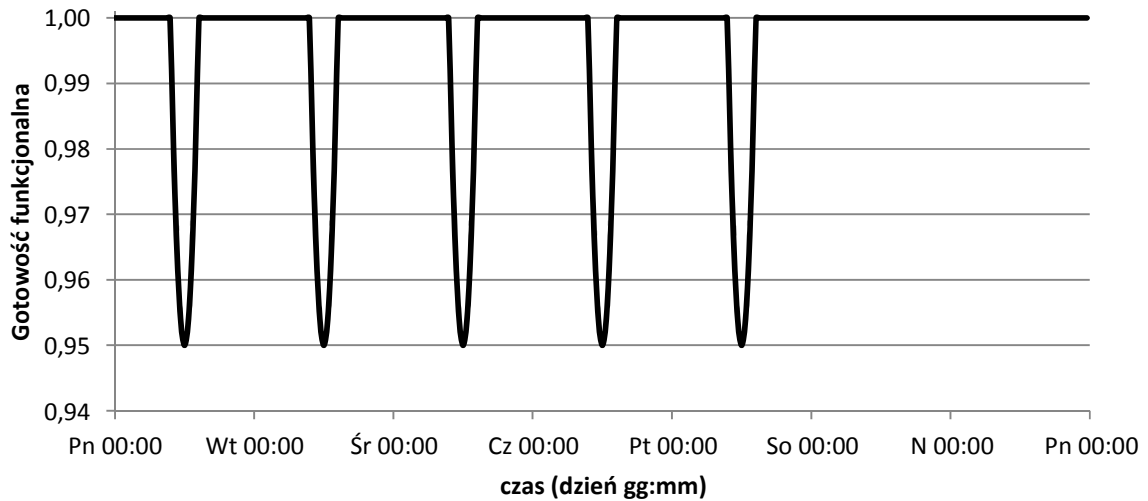
Przyjmując tygodniową zmienność ruchu przedstawioną na Rys. 3 można wyznaczyć zmienność wartości gotowości funkcjonalnej w czasie (Rys. 4).



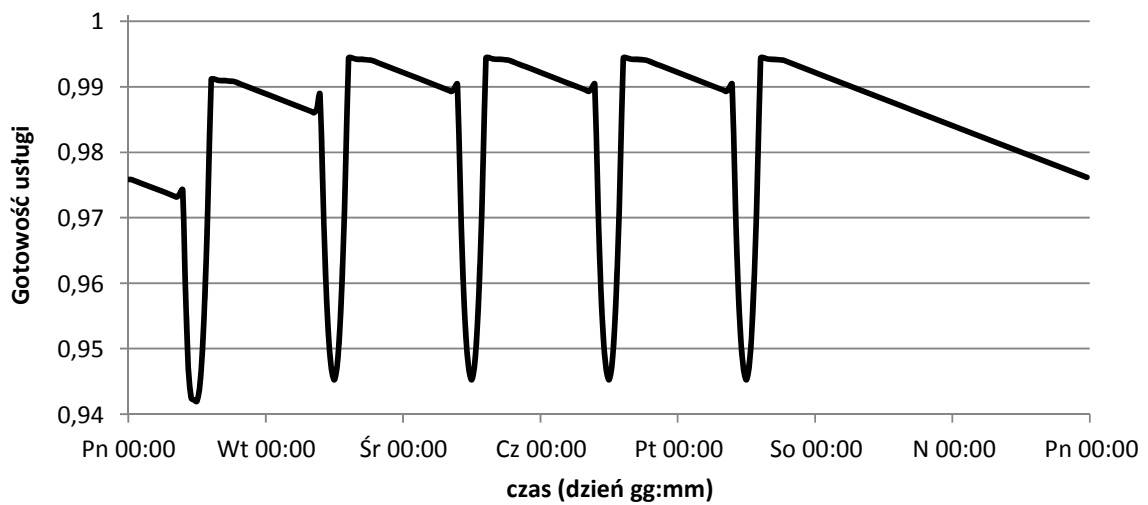
Rys. 2. Gotowość funkcjonalna w funkcji intensywności zgłoszeń dla przykładowego systemu



Rys. 3. Zmiany intensywności zgłoszeń dla przykładowego systemu w okresie tygodniowym



Rys. 4. Gotowość funkcjonalna w funkcji intensywności zgłoszeń dla przykładowego systemu



Rys. 5. Gotowość usługi w ciągu tygodnia dla przykładowego systemu

6.3. Gotowość usługi

Zgodnie z zależnością (3) iloczyn gotowości systemu (Rys. 1) i funkcjonalnej (Rys. 4) pozwala wyznaczyć gotowość usługi dla przykładowego systemu (Rys. 5). Można zauważyć dobową zmienność wartości gotowości wynikającą z dobowej zmienności obciążenia systemu oraz godzin pracy administratorów. Ponadto wyraźny jest spadek gotowości w weekend osiągający swoje minimum w poniedziałek około południa. Wartość minimalna gotowości w poniedziałek związana jest z kumulacją uszkodzeń z soboty i niedzieli i dobowym maksimum napływu użytkowników.

7. Uwagi końcowe

7.1 Podsumowanie

W artykule przedstawiono metodę wyznaczania zmienności gotowości usług internetowych w ciągu tygodnia. Metoda bierze pod uwagę aspekty niezawodnościowe i funkcjonalne. Analiza niezawodnościowa uwzględniająca prawdopodobieństwo, że system będzie zdalny w danym momencie czasu bierze pod uwagę strukturę systemu, losowe czasy do uszkodzeń (sprzętu i oprogramowania – w skutek naruszenia zabezpieczeń) i realistyczny model odnowy biorący pod uwagę godziny pracy administratorów systemu i czas związany z dostarczeniem nowych części. Analiza funkcjonalna pozwala na wyznaczanie prawdopodobieństwa, że użytkownik zostanie obsłużony w czasie mniejszym niż zadany. Bierze ona pod uwagę zmienność intensywności napływu użytkowników oraz parametry techniczne usługi: jej choreografię, alokację zadań na węzły, parametry techniczne węzłów i zadań. Obliczenia numeryczne wykonane zostały przy pomocy opracowanych przez autora symulatorów komputerowych. Wykorzystanie techniki symulacji Monte-Carlo pozwoliło na stworzenie elastycznego modelu niezawodnościowo-funkcjonalnego systemu usług internetowych. Posiada ono jednak kilka istotnych wad. Po pierwsze czas symulacji związany z wielokrotnymi powtórzeniami (podstawowa zasada symulacji Monte-Carlo) wymaga dużych zasobów komputerowych. Po drugie zmiany w strukturze modelu (ale nie w jego parametrach) mogą wymagać zmian w kodzie źródłowym symulatora. Przedstawione podejście pozwala w łatwy sposób zbadać wpływ zmian w systemie napraw systemu komputerowego (np. godzin pracy administratorów) czy też zmian w parametrach funkcjonalnych systemu (np. innej alokacji zadań, zwiększania poziomu zabezpieczeń czy wymiany węzłów na wydajniejsze) na istotny z punktu widzenia użytkownika usługi parametr: gotowość.

7.2 Plan dalszych prac

Przedstawiona metoda zawiera istotne założenia ograniczające jej zakres stosowania. Zakłada, że uszkodzenia sprzętu komputerowego czy oprogramowania objawiają się niezdatnością całego systemu. Jest to prawdą dla systemów internetowych, które nie wykorzystują techniki równoważenia obciążeń czy przetwarzania w chmurze. Zatem uszkodzenie jednego z komputerów nie blokuje działania całego systemu, objawia się jedynie zmniejszeniem jego wydajności, w efekcie następuje wydłużenie czasu obsługi użytkownika. Drugim założeniem, jest nieuwzględnianie zdarzeń objawiających się zmniejszeniem wydajności węzła, takich jak ataki czy wirusy. Możliwe jest rozszerzenie prezentowanego podejścia o omawiane aspekty poprzez zwiększenie liczby analizowanych stanów niezawodnościowych ponad dwa omawiane w artykule w rozdziale 5 (zdalny i niezdatny). Każdy stan systemu opisany będzie przez stan każdego z jego elementów (serwerów). Każdy element systemu może być w jednym z trzech stanów: zdalny, uszkodzenie oraz ograniczenie wydajności. Dla każdego z N stanów (oznaczymy je jako S_i) można wyznaczyć prawdopodobieństwo (oznaczymy je jako $\Pr(S_i, t)$) przebywania systemu w danym stanie

poprzez rozbudowę symulatora omawianego w rozdziale 4. Wykorzystując podejście i symulator opisany w rozdziale 5 można wyznaczyć gotowość funkcjonalną w danym stanie ($\Pr(\text{czasodpowiedzi}(T) < t_{\max} | T = t \wedge S_i)$) i w efekcie otrzymać wartości funkcji gotowości usługi:

$$A(t) = \sum_{i=1}^N \Pr(\text{czasodpowiedzi}(T) < t_{\max} | T = t \wedge S_i) \cdot \Pr(S_i, t). \quad (10)$$

Podstawowym problem takiego rozwiązania jest liczba stanów rosnąca wykładniczo, z wykładnikiem równym dwukrotnej liczbie użytych komputerów [28]. Daje to ponad 4096 stanów dla systemu składającego się z 6 węzłów (omawianego w niniejszym rozdziale). Dla każdego, ze stanów należy przeprowadzić czasochłonną symulację funkcjonalną (rozdział 4). Rozwiązaniem tego problemu jest ograniczenie sumy we wzorze (10) do stanów o największym prawdopodobieństwie wystąpienia i odrzucenie stanów najmniej prawdopodobnych.

Kolejnym planowanym obszarem prac jest zastosowanie zaproponowanej metody do analizy usług internetowych realizowanych w chmurze obliczeniowej. W takim przypadku aspekt uszkodzeń sprzętowych pełnił będzie dużo mniejszą rolę. Jest to efekt stosowanie technik wirtualizacji i migracji maszyn wirtualnych w chmurze w przypadku uszkodzenia jednego z serwerów. Jednakże uszkodzenia oprogramowania występujące najczęściej w skutek naruszenia zabezpieczeń jak i proces odnowy systemu po takim uszkodzeniu może być modelowany w sposób analogiczny do przedstawionego w pracy.

Praca naukowa finansowana ze środków na naukę w latach 2011-2014 jako projekt badawczy. Nr projektu N N516 475940.

Literatura

1. Aven T, Uwe J. Stochastic Models in Reliability. New York: Springer, 1999.
2. Banerjee S, Srikanth H, Cukic B. Log-based reliability analysis of Software as a Service (SaaS). Proceedings of the 21st International Symposium on Software Reliability Engineering 2010; 1: 239-248.
3. Barlow R, Proschan F. Mathematical Theory of Reliability. Philadelphia: Society for Industrial and Applied Mathematics, 1996.
4. Birta L, Arbez G. Modelling and Simulation: Exploring Dynamic System Behaviour. London: Springer, 2007.
5. Clymer J. Simulation-based engineering of complex systems. Wiley Series in Systems Engineering and Management, Wiley-Interscience, 2009.
6. Faulin J, Juan A, Serrat C, Bargueño V. Predicting availability functions in time-dependent complex systems with SAEDES simulation algorithms. Reliability Engineering & System Safety, 2008; 93(11): 1761-1771.
7. Finkelstein M. A scale model of general repair, Microelectron. Reliab, 1993; 33(1): 41-44.
8. Fishman G. Monte Carlo: Concepts, Algorithms, and Applications, Springer-Verlag, 1996.

9. Gokhale S S, Jijun L. Performance and Availability Analysis of an E-Commerce Site. *Computer Software and Applications Conference*, 2006; 1: 495-502.
10. Gold N, Knight C, Mohan A, Munro M. Understanding service-oriented software. *IEEE Software* 2004; 21: 71-77.
11. Goseva-Popstojanova K, Singh A D, Mazimdar S, Li F. Empirical characterization of session-based workload and reliability for Web servers. *Empirical Software Engineering*; 11(1): 71-117.
12. Janevski N, Goseva-Popstojanova K. Session Reliability of Web Systems Under Heavy-Tailed Workloads: An Approach based on Design and Analysis of Experiments. *IEEE Transactions on Software Engineering*, 2013; 99(PrePrints): 1.
13. Jia J, Wu S. A replacement policy for a repairable system with its repairman having multiple vacations. *Computers & Industrial Engineering*, 2009; 57(1): 156-160.
14. Jishen J, Shaomin W. Optimizing replacement policy for a cold-standby system with waiting repair times. *Applied Mathematics and Computation Journal*, 2009; 133-141.
15. Juan A A, Faulin J, Ramirez-Marquez J E, Martorell Alsina S S. *Simulation Methods for Reliability and Availability of Complex Systems*, Springer 2012.
16. Lam Y. A note on the optimal replacement problem, *Adv. Appl. Prob.* 1998; 20(2): 479-482.
17. Lavenberg S. A perspective on queueing models of computer performance. *Performance Evaluation*, 1989; 10:53-76.
18. Lipinski Z. State Model of Service Reliability. *International Conference on Dependability of Computer Systems*. IEEE Computer Society 2006; 35-42.
19. Liu X, Jin H, Sha L, Zhu X. Adaptive control of multi-tiered Web application using queueing predictor. *Proc. IEEE/IFIP Netw. Operations Management Symp. (NOMS2006)*, 2006; 106-114.
20. Liu J. Parallel Real-time Immersive Modelling Environment (PRIME), Scalable Simulation Framework (SSF), User's manual, Colorado School of Mines Department of Mathematical and Computer Sciences. [Available online: <http://prime.mines.edu/>], 2006.
21. Malinowski J. A simulation model for complex repairable systems with inter-component dependencies and three types of component failures. *Advances in Safety, Reliability and Risk Management - Proceedings of the European Safety and Reliability Conference, ESREL 2011, 2012*; 128.
22. Merzbacher M, Patterson D. Measuring end-user availability on the Web: practical experience. *Proceedings of the International Conference on Dependable Systems and Networks*, 2002; 473 – 477.
23. Michalska K, Walkowiak T. Simulation approach to performance analysis information systems with load balancer. *Information systems architecture and*

technology: advances in Web-Age Information Systems, Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej 2009; 269-278.

24. Rahmawan H, Gondokaryono Y S. The simulation of static load balancing algorithms. International Conference on Electrical Engineering and Informatics, 2009; 640-645.
25. Urgaonkar B, Pacificiy G, Shenoy P, Spreitzery M, Tantawi A. An analytical model for multitier internet services and its applications. Proc. of the ACM SIGMETRICS'05 Conference on measurement and modeling of computer systems, 2005; 291-302.
26. Walkowiak T. Information systems performance analysis using task-level simulator. International Conference on Dependability of Computer Systems. IEEE Computer Society Press, 2009; 218-225.
27. Walkowiak T. Simulation approach to Web system dependability analysis. Summer Safety and Reliability Seminars, SSARS 2011; 1: 197-204.
28. Walkowiak T, Michalska K. Functional based reliability analysis of Web based information systems. Dependable computer systems / Wojciech Zamojski [et al.] (eds.). Berlin ; Heidelberg: Springer, 2011; 257-269.
29. Zamojski W. Redundancja sprzętowa współczesnych systemów technicznych. Nadmiarowość w inżynierii niezawodności. XXXII Zimowa Szkoła Niezawodności, Szczyrk. Radom: Instytut Technologii Eksploatacji 2004; 398-411.
30. Zhang Y L, Wu S. Reliability analysis for a k/n(F) system with repairable repair-equipment, Appl. Math. Model. 2009; 33(7): 3052-3067.