

## **SwinDeW-C: system przepływu pracy w chmurze oparty na peer-to-peer**

### **Wstęp**

Systemy przepływu pracy mają na celu wsparcie automatyzacji procesów w aplikacjach biznesowych i naukowych na dużą skalę. W ostatnich latach wiele systemów przepływu pracy zostało wdrożonych w infrastrukturach obliczeniowych o wysokiej wydajności, takich jak klastry, peer-to-peer (p2p) i przetwarzanie siatkowe. Jedną z sił napędowych jest rosnące zapotrzebowanie na instancje na dużą skalę i aplikacje przepływu pracy wymagające dużej ilości danych/obliczeń (w skrócie aplikacje przepływu pracy na dużą skalę), które są powszechne zarówno w obszarach aplikacji e-biznesowych, jak i eScience. Typowe przykłady obejmują takie, jak ogólnokrajowy proces składania wniosków o odszkodowanie z tytułu intensywnej transakcji; intensywny proces wyszukiwania pulsarów w astrofizyce. Mówiąc ogólnie, aplikacje intensywnie korzystające z instancji to te procesy, które muszą być wykonywane wiele razy sekwencyjnie w bardzo krótkim czasie lub jednocześnie z dużą liczbą instancji. W związku z tym aplikacje przepływu pracy na dużą skalę zwykle wymagają obsługi infrastruktur obliczeniowych o wysokiej wydajności (np. zaawansowanych jednostek procesora, dużej przestrzeni pamięci i szybkiej sieci), zwłaszcza gdy działania przepływu pracy same w sobie wymagają dużej ilości danych i obliczeń. W rzeczywistym świecie, aby sprostać takim wymaganiom, drogie infrastruktury obliczeniowe, w tym superkomputery i serwery danych, są kupowane, instalowane, integrowane i utrzymywane przy ogromnych kosztach przez użytkowników systemu. Ponieważ jednak większość tych zasobów jest samowystarczalna i zorganizowana w sposób heterogeniczny, skalowalność zasobów, tj. to, jak łatwo system może rozszerzać i kurczyć swoją pulę zasobów, aby obsłużyć cięższe lub lżejsze obciążenia pracą, jest bardzo słaba. Z powodu takiego problemu, z jednej strony, pozyskiwanie osób z zewnątrz wiąże się z dużymi, jeśli nie niemożliwymi, kosztami zasobów na rozwiązanie problemu „niedoboru zasobów” w okresach szczytu; z drugiej strony nie może świadczyć usług na rzecz innych w okresach poza szczytem, aby w pełni wykorzystała inwestycję. W obecnych paradygmatach obliczeniowych systemy przepływu pracy muszą utrzymywać własne infrastruktury obliczeniowe o wysokiej wydajności, zamiast wykorzystywać je jako usługi od strony trzeciej zgodnie z ich rzeczywistymi potrzebami. Tymczasem większość zasobów jest bezczynna, z wyjątkiem dużych wymagań dotyczących zasobów w aplikacjach przepływu pracy na dużą skalę w okresach szczytu. W rzeczywistości wiele systemów przepływu pracy musi również przez większość czasu radzić sobie z dużą liczbą konwencjonalnych, mniej wymagających aplikacji przepływu pracy. Dlatego skalowalność zasobów staje się krytycznym problemem dla obecnych systemów przepływu pracy. Jednak taki problem nie został dobrze rozwiązany przez obecne paradygmaty obliczeniowe, takie jak przetwarzanie klastrowe i siatkowe. W ostatnich latach przetwarzanie w chmurze staje się najnowszym paradygmatem przetwarzania rozproszonego i cieszy się coraz większym zainteresowaniem badaczy z obszaru Przetwarzania Rozproszonego i Równoległego, Informatyki Zorientowanej na Usługi oraz Inżynierii Oprogramowania. Jak zaproponował Ian Foster i podzieliła to wielu badaczy i praktyków, w porównaniu z konwencjonalnymi paradygmatami obliczeniowymi, przetwarzanie w chmurze może zapewnić „pulę wyabstrahowanych, zwirtualizowanych, dynamicznie skalowalnych, zarządzanych zasilenie, pamięć masowa, platformy i usługi są dostarczane na żądanie klientom zewnętrznym przez Internet”. Dlatego przetwarzanie w chmurze może zapewnić skalowalne zasoby na żądanie do wymagań systemu. Tymczasem przetwarzanie w chmurze przyjmuje model biznesowy zorientowany na rynek, w którym użytkownicy są obciążani opłatami za korzystanie z usług w chmurze, takich jak usługi obliczeniowe, przechowywanie i usługi sieciowe, takie jak konwencjonalne media w życiu codziennym (np. woda, prąd, gaz i telefonia). Oczywiście możliwe jest wykorzystanie przetwarzania w chmurze do rozwiązania problemu skalowalności zasobów do zarządzania aplikacjami przepływu pracy na dużą skalę. Dlatego badanie systemów workflow opartych na chmurze obliczeniowej, czyli systemów workflow w chmurze, jest kwestią aktualną i opłacalną dla zwiększenia wysiłków. Oprócz

skalowalnych zasobów, kolejną zasadniczą kwestią związaną z aplikacjami przepływu pracy na dużą skalę jest zdecentralizowane zarządzanie. W celu osiągnięcia pomyslanej realizacji, wymagana jest efektywna koordynacja uczestników systemu (np. dostawców usług, odbiorców usług i brokerów usług) dla wielu zadań zarządzania, takich jak zarządzanie zasobami (zarządzanie obciążeniem, harmonogramowanie przepływu pracy), zarządzanie QoS (Quality of Service), zarządzanie danymi zarządzanie, zarządzanie bezpieczeństwem i inne. Jednym z konwencjonalnych sposobów rozwiązania problemu koordynacji jest scentralizowane zarządzanie, w którym usługi koordynacji są konfigurowane na scentralizowanej maszynie. Cała komunikacja, taka jak dane i komunikaty kontrolne, jest przesyłana tylko między węzłem centralnym a innymi węzłami zasobów, ale nie między nimi. Jednak scentralizowane zarządzanie zależy w dużej mierze od węzła centralnego i dlatego może łatwo spowodować wąskie gardło wydajności. Inne typowe wady to również: pojedynczy punkt awarii, brak skalowalności i zaawansowana moc obliczeniowa wymagana do usług koordynacji. Aby przezwyciężyć problemy zarządzania scentralizowanego, zarządzania zdecentralizowanego, w którym scentralizowane repozytorium danych i silnik kontrolny są porzucane, a zarówno dane, jak i komunikaty kontrolne są przesyłane między wszystkimi węzłami za pośrednictwem ogólnych mechanizmów transmisji rozgłoszeniowej lub ograniczonych mechanizmów komunikacji rozgłoszeniowej. W ten sposób wąskie gardła wydajności są prawdopodobnie wyeliminowane, a skalowalność systemu może zostać znacznie zwiększona. Peer to Peer (p2p) to typowa zdecentralizowana architektura. Jednak bez jakiegokolwiek scentralizowanej koordynacji, czyste p2p (nieustrukturyzowane zdecentralizowane), w którym wszystkie węzły równorzędne komunikują się ze sobą poprzez pełne rozgłoszanie, cierpi z powodu niskiej wydajności i dużego obciążenia sieci. Najwyraźniej ani scentralizowane, ani nieustrukturyzowane zarządzanie zdecentralizowane nie nadaje się do zarządzania na dużą skalę aplikacji przepływu pracy, ponieważ wymagane są masowe usługi komunikacyjne i koordynacyjne. Dlatego w praktyce często stosuje się ustrukturyzowaną architekturę p2p, w której superwęzeł działa jako równorzędni koordynatorzy dla grupy równorzędnych. Dzięki tym superwęzłom, które przechowują wszystkie niezbędne informacje o sąsiednich węzłach, zadania zarządzania przepływem pracy mogą być skutecznie wykonywane, gdy dane i komunikaty kontrolne są transmitowane w ograniczony sposób. Dlatego też zdecentralizowane zarządzanie strukturalne jest skuteczniejsze niż inne w zarządzaniu aplikacjami przepływu pracy. Na podstawie powyższej analizy oczywiste jest, że przetwarzanie w chmurze jest obiecującym rozwiązaniem spełniającym wymagania dotyczące skalowalnych zasobów, a ustrukturyzowana zdecentralizowana architektura, taka jak ustrukturyzowane p2p, jest skutecznym rozwiązaniem spełniającym wymóg zdecentralizowanego zarządzania. Dlatego w tym rozdziale przedstawiamy SwinDeW-C (Swinburne Decentralized Workflow for Cloud), oparty na peer to peer system przepływu pracy w chmurze do zarządzania aplikacjami przepływu pracy na dużą skalę. SwinDeW-C nie jest budowany od zera, ale oparty na naszym istniejącym SwinDeW-G (systemie przepływu pracy opartym na sieci peer-to-peer), który zostanie przedstawiony w dalszej części. Zgodnie z ustaleniami wielu badaczy i praktyków ogólna architektura chmury obejmuje cztery podstawowe warstwy od góry do dołu: warstwę aplikacji (aplikacje użytkownika), warstwę platformy (usługi w chmurze oprogramowania pośredniczącego ułatwiające tworzenie/wdrażanie aplikacji użytkownika), warstwę zunifikowanych zasobów (abstrakt /enkapsulowane zasoby przez wirtualizację) i warstwę szkieletową (fizyczne zasoby sprzętowe). Aby obsługiwać aplikacje przepływu pracy na dużą skalę, zaprezentowano nowatorską architekturę SwinDeW-C, w której oryginalna warstwa tkaniny SwinDeW-G jest dziedziczona z rozszerzeniem zewnętrznych komercyjnych dostawców usług w chmurze. W międzyczasie dokonano znaczących modyfikacji w pozostałych trzech warstwach: podstawowe zasoby są zwirtualizowane w zunifikowanej warstwie zasobów; elementy funkcjonalne są dodawane lub ulepszone w warstwie platformy, aby wspierać zarządzanie aplikacjami przepływu pracy na dużą skalę; interfejs użytkownika został zmodyfikowany w celu obsługi dostępu do Internetu (przeglądarki internetowej). Tu opisano

nowatorską architekturę systemu i nowe funkcje SwinDeW-C. W szczególności, w oparciu o krótkie wprowadzenie o SwinDeW-G, zaproponowano architekturę SwinDeW-C, a także architekturę elementów równorzędnych SwinDeW-C (w tym zarówno zwykłych elementów równorzędnych, jak i elementów równorzędnych koordynatora). Tymczasem oprócz wspólnych funkcji dla systemów przetwarzania w chmurze i systemów przepływu pracy, dodatkowe nowe komponenty funkcjonalne są ulepszone lub projektowane w SwinDeW-C w celu ułatwienia aplikacji przepływu pracy na dużą skalę. Tu, trzy nowe elementy funkcjonalne, w tym zarządzanie QoS, zarządzanie danymi i zarządzanie bezpieczeństwem, przedstawiono jako kluczowe elementy do zarządzania aplikacjami przepływu pracy na dużą skalę. Zademonstrowano prototypowy system SwinDeW-C w celu weryfikacji efektywności architektury SwinDeW-C oraz możliwości zbudowania systemu workflow w chmurze w oparciu o istniejącą platformę grid computing.

### **Motywacja i wymagania systemowe**

W tej sekcji najpierw przedstawimy kilka przykładów ilustrujących motywację do wykorzystania przetwarzania w chmurze w celu ułatwienia aplikacji przepływu pracy na dużą skalę. Następnie, w oparciu o wprowadzenie naszego istniejącego, gridowego środowiska obliczeniowego SwinDeW-G, przedstawiono wymagania systemowe dla systemów workflow w chmurze.

### **Aplikacje przepływu pracy na dużą skalę**

Poniżej przedstawiamy dwa przykłady, jeden pochodzi z obszaru zastosowań biznesowych (roszczenie ubezpieczeniowe), a drugi z obszaru zastosowań naukowych (wyszukiwanie pulsarowe). Roszczenie ubezpieczeniowe: Proces roszczenia ubezpieczeniowego to typowy biznesowy przepływ pracy, który zapewnia usługi dla procesów takich jak ubezpieczenie w ramach świadczeń pracowniczych, w tym na przykład koszty leczenia, emerytury i zasiłki dla bezrobotnych. Ze względu na rozproszone lokalizacje geograficzne dużej liczby wnioskodawców, biura ubezpieczeniowe są zwykle rozmieszczone w wielu lokalizacjach na dużym obszarze, obsługując ogromną populację. Pomimo różnic między konkretnymi aplikacjami, w dużych/średnich firmach ubezpieczeniowych często występują następujące wymagania: (1) obsługa dużej liczby procesów wywoływanych w bezpieczny sposób z dowolnego miejsca w Internecie, prywatność wnioskodawców i poufne dane muszą być chronione; (2) unikanie zarządzania systemem w wielu różnych lokalizacjach ze względu na wysokie koszty ustawienia i bieżącej konserwacji; (3) możliwość obsługi ogromnej populacji obejmującej procesy w minimalnej skali kilkudziesięciu tysięcy dziennie, czyli np. intensywne; oraz (4) dla lepszej jakości usług, konieczność odpowiedniej obsługi wyjątków przepływu pracy, szczególnie w przypadku procesów o dużej liczbie instancji. Wyszukiwanie pulsarowe: Proces wyszukiwania pulsarów jest typowym przepływem pracy naukowej, który obejmuje dużą liczbę czynności wymagających dużej ilości danych i obliczeń. W przypadku pojedynczego procesu wyszukiwania, średnia ilość danych (nie licząc nieprzetworzonych danych strumienia z teleskopu) może przekraczać terabajty, a średni czas wykonania może wynosić około jednego dnia. W pojedynczym procesie wyszukiwania należy wykonać wiele równoległych ścieżek wykonania dla danych zebranych z różnych wiązek teleskopu, a każda ścieżka wykonania obejmuje cztery główne segmenty: zbieranie danych, wstępne przetwarzanie danych, wyszukiwanie kandydatów i podejmowanie decyzji. Jako przykład weźmy operację dedypersji w segmencie wstępnego przetwarzania danych. Dedypersja ma na celu wygenerowanie dużej liczby plików dedypersji w celu skorygowania sygnałów pulsarowych, które są rozpraszane przez ośrodek międzygwiazdowy. Duża liczba plików dedypersji musi być generowana zgodnie z różnymi wyborami rozproszenia próbnego i zwykle zajmuje wiele godzin na zasobach obliczeniowych o wysokiej wydajności. Po wygenerowaniu dużej ilości plików dedypersji, zostaną wdrożone różne algorytmy wyszukiwania kandydatów na pulsary, takie jak wyszukiwanie FFT, wyszukiwanie FFA i wyszukiwanie pojedynczego impulsu. Na koniec wyniki zostaną zwizualizowane, aby wesprzeć decyzję ekspertów-

człowieka, czy znaleziono pulsar, czy nie. Ogólnie rzecz biorąc, proces wyszukiwania pulsarów często stawia następujące wymagania: (1) łatwość skalowania w górę i w dół wykorzystywanych zasobów obliczeniowych do przetwarzania danych na różnych etapach; (2) dla lepszej QoS, szczególnie wydajnego planowania zadań obliczeń równoległych, tak aby każdy proces wyszukiwania pulsarów mógł zostać ukończony na czas; (3) zmniejszenie kosztów przechowywania i przesyłania danych, wymagane są określone strategie w celu określenia alokacji wygenerowanych danych podczas wykonywania przepływu pracy; (4) wybór godnych zaufania węzłów usługowych, zapewniających bezpieczeństwo przechowywania danych, zwłaszcza tych, które muszą być przechowywane przez długi czas.

### **Wymagania systemowe**

Na podstawie wstępu o powyższych dwóch przykładach przedstawiamy tutaj wymagania systemowe do zarządzania aplikacjami przepływu pracy na dużą skalę. Oprócz dwóch podstawowych wymagań, a mianowicie skalowalnych zasobów i zdecentralizowanego zarządzania, które zostały omówione we wstępie, istnieją trzy ważne wymagania systemowe, w tym: zarządzanie QoS, zarządzanie danymi i zarządzanie bezpieczeństwem.

### **Zarządzanie QoS**

Kluczowe znaczenie ma dostarczanie usług z jakością usług (QoS) usatysfakcjonowaną przez użytkowników w środowisku chmury obliczeniowej, w przeciwnym razie reputacja dostawców usług ulegnie pogorszeniu i ostatecznie zostanie wyeliminowana z rynku chmury. Ogólnie rzecz biorąc, istnieje 5 głównych wymiarów ograniczeń QoS, w tym czas, koszt, wierność, niezawodność i bezpieczeństwo. Wśród nich czas, jako podstawowy miernik wydajności systemu, jest prawdopodobnie najbardziej ogólnym ograniczeniem QoS we wszystkich systemach aplikacyjnych. Szczególnie w przypadku aplikacji przepływu pracy na dużą skalę, czasowa QoS jest bardzo ważna, ponieważ wszelkie duże opóźnienia mogą skutkować niską wydajnością, a nawet wąskimi gardłami systemu. Dlatego w tym artykule skupiamy się głównie na ograniczeniach czasowych jako przykładzie zarządzania QoS. W przypadku ogólnej aplikacji przepływu pracy w chmurze zarówno globalne, jak i lokalne ograniczenia czasowe są przypisywane w czasie kompilacji przepływu pracy poprzez umowę dotyczącą poziomu usług (SLA). Następnie w czasie wykonywania przepływu pracy, ze względu na bardzo dynamiczną wydajność systemu (czas trwania czynności z dużymi odchyleniami (Liu, Chen, Liu i Yang, 2008)), wykonanie przepływu pracy jest stale monitorowane pod kątem naruszeń tych ograniczeń czasowych (Chen i Yang, 2008a, 2010, 2008b). Jeśli zostanie wykryte tymczasowe naruszenie (tj. czas wykonania przepływu pracy przekroczy ograniczenie czasowe), zostaną uruchomione strategie obsługi wyjątków w celu skompensowania występujących opóźnień czasowych. W związku z tym, aby zapewnić satysfakcjonującą czasową QoS (a także inne ograniczenia QoS), należy przyjąć lub zaprojektować zestaw strategii, aby ułatwić co najmniej trzy następujące zadania: ustawienie ograniczeń QoS, monitorowanie wykonania przepływu pracy pod kątem naruszeń ograniczeń QoS, oraz obsługa naruszeń ograniczeń QoS.

### **Zarządzanie danymi**

Aplikacje przepływu pracy na dużą skalę często towarzyszą intensywnemu przetwarzaniu danych, gdzie zadania przepływu pracy będą uzyskiwać dostęp do dużych zbiorów danych w celu zapytania lub pobrania danych, a podczas wykonywania przepływu pracy będą generowane jako produkty pośrednie lub końcowe. Zarządzanie danymi w systemach workflow w chmurze ma nowe wymagania, co staje się istotną kwestią. Po pierwsze, nowa strategia przechowywania danych jest wymagana w systemach workflow w chmurze. W chmurze obliczeniowej teoretycznie system może oferować nieograniczone zasoby pamięci masowej. Wszystkie dane aplikacji mogą być przechowywane, w tym dane pośrednie,

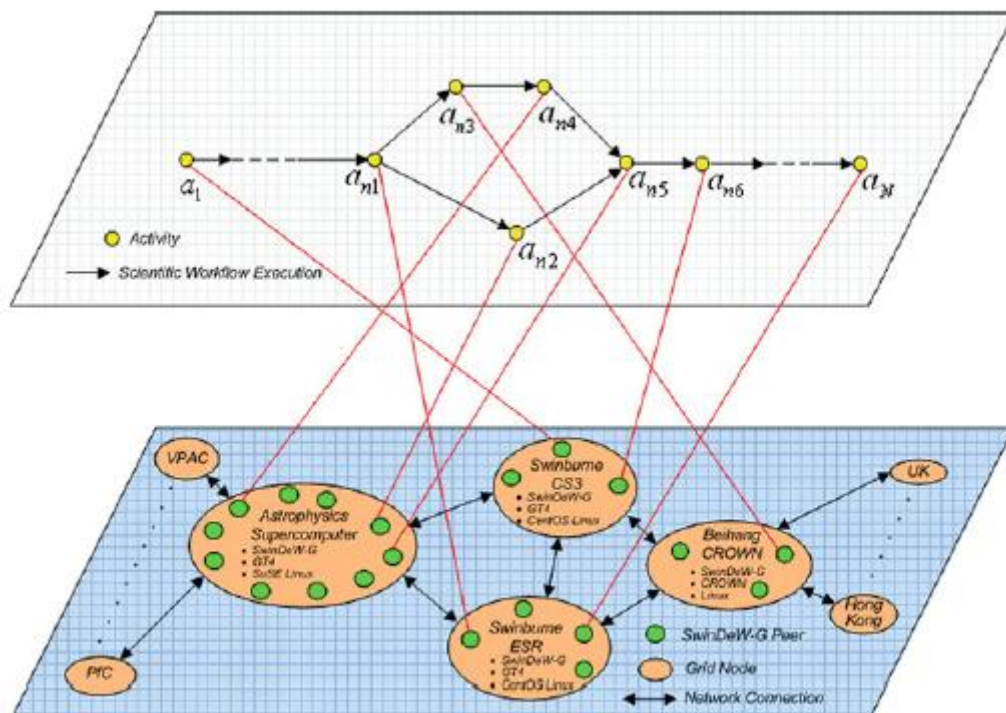
jeśli jesteśmy gotowi zapłacić za wymagane zasoby. Dlatego potrzebujemy strategii opłacalnego przechowywania dużych danych aplikacji. Po drugie, wymagana jest również nowa strategia umieszczania danych. Platforma przetwarzania w chmurze obejmuje różnych dostawców usług w chmurze z różnymi modelami cenowymi, w przypadku których transfer danych między dostawcami usług również wiąże się z kosztami. Przepływy pracy w chmurze są zwykle rozproszone, a strategia rozmieszczania danych zdecyduje, gdzie przechowywać dane aplikacji, aby zmniejszyć całkowity koszt systemu. Wreszcie, nowa strategia replikacji danych powinna być również zaprojektowana dla systemów przepływu pracy w chmurze. Dobra strategia replikacji może nie tylko zagwarantować bezpieczeństwo danych aplikacji, ale także dodatkowo obniżyć koszty systemu poprzez replikację często używanych danych w różnych lokalizacjach. Strategia replikacji w chmurze powinna być dynamiczna w oparciu o wskaźnik wykorzystania danych aplikacji.

### **Zarządzanie bezpieczeństwem**

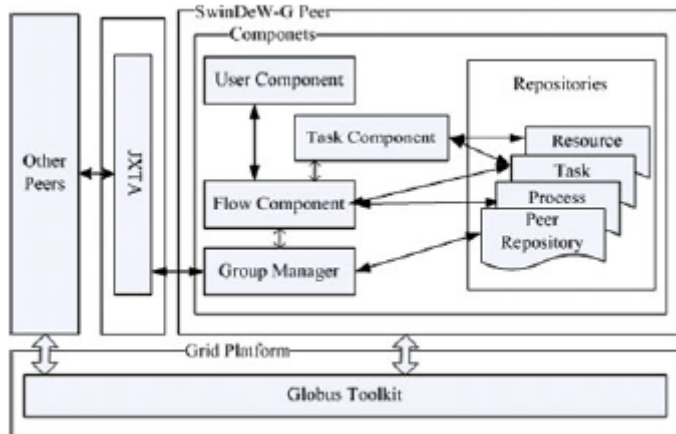
Bezpieczeństwo zawsze odgrywa ważną rolę w rozproszonych systemach obliczeniowych. Aby zapewnić wysoki QoS tych systemów, skupiamy się na problemach bezpieczeństwa, jakie niosą ze sobą różne typy komponentów, duża ilość heterogenicznych danych oraz nieprzewidywalne procesy wykonawcze. Ponieważ niektóre ogólne aspekty bezpieczeństwa systemu, takie jak jakość usług i bezpieczeństwo danych, są częściowo uwzględnione w poprzednich komponentach QoS i zarządzania danymi, ten rozdział kładzie nacisk na zarządzanie zaufaniem, które odgrywa ważną rolę w zarządzaniu peerami SwinDeW-C. W aplikacjach przepływu pracy na dużą skalę, aby sprostać wysokim wymaganiom jakości i skalowalności, wydajne i adaptacyjne zarządzanie zaufaniem jest nieodzowną częścią platformy SwinDeW-C. Poza tym zarządzanie użytkownikami jest niezbędne, aby zagwarantować bezpieczeństwo systemu i uniknąć nielegalnego użytkownika. W obliczu skomplikowanych struktur sieciowych w środowisku chmury potrzebujemy również technologii szyfrowania, aby chronić prywatność, integralność, autentyczność i niezaprzeczalność. Biorąc pod uwagę te podstawowe wymagania systemowe, przetwarzanie w chmurze jest odpowiednim rozwiązaniem problemu skalowalności zasobów, a p2p jest skutecznym kandydatem do zarządzania zdecentralizowanego. Tymczasem, aby dostosować wymagania systemowe aplikacji przepływu pracy na dużą skalę, elementy funkcjonalne do zarządzania danymi, zarządzania QoS i zarządzania bezpieczeństwem muszą zostać zaprojektowane lub ulepszone, aby zagwarantować zadowalającą wydajność systemu.

### **Przegląd środowiska SwinDeW-G**

Zanim przedstawimy SwinDeW-C, należy przedstawić trochę podstawowej wiedzy na temat SwinDeW-G. SwinDeW-G (Swinburne Decentralised Workflow for Grid) to naukowy system przepływu pracy oparty na sieci peer-to-peer, działający na platformie SwinGrid (Swinburne service Grid). Ogólny obraz SwinGrid przedstawiono na rysunku (dolna płaszczyzna).



SwinGrid zawiera wiele węzłów siatki rozmieszczonych w różnych miejscach. Każdy węzeł sieci zawiera wiele komputerów, w tym wysokowydajne komputery PC i/lub superkomputery złożone ze znacznej liczby jednostek obliczeniowych. Główne węzły hostingowe obejmują węzeł Swinburne CS3 (centrum złożonych systemów i usług oprogramowania), węzeł Swinburne ESR (laboratorium badań systemów przedsiębiorstwa), węzeł superkomputera Swinburne Astrophysics oraz węzeł Beihang CROWN (chińskie środowisko badawczo-rozwojowe Over Wide-area Network). Węzeł w Chinach. Używają Linuksa, GT4 (Globus Toolkit) lub CROWN grid toolkit 2.5, gdzie CROWN jest rozszerzeniem GT4 z większą ilością oprogramowania pośredniego, a zatem jest kompatybilny z GT4. Węzeł CROWN jest również połączony z kilkoma innymi węzłami, takimi jak na Uniwersytecie Nauki i Technologii w Hongkongu oraz na Uniwersytecie w Leeds w Wielkiej Brytanii. Swinburne Astrophysics Supercomputer Node współpracuje z australijskim Pfc (Platforms for Collaboration) i VPAC (Victorian Partnership for Advanced Computing). Obecnie SwinDeW-G jest wdrożony we wszystkich głównych węzłach hostingowych, jak pokazano w górnej płaszczyźnie. W SwinDeW-G przepływ pracy naukowej jest wykonywany przez różnych partnerów, które mogą być rozmieszczone w różnych węzłach sieci. Jak pokazano na rysunku, każdy węzeł sieci może mieć pewną liczbę węzłów, a każdy węzeł może być po prostu oglądany jako usługa sieci. W górnej płaszczyźnie pokazujemy próbkę tego, jak naukowy przepływ pracy może być wykonany w środowisku przetwarzania siatkowego. Podstawową jednostką usługową w SwinDeW-G jest peer SwinDeW-G, który działa jako usługa sieciowa wraz z innymi usługami sieciowymi. Jednak komunikuje się z innymi peerami poprzez JXTA (<http://www.sun.com/software/jxta/>), platformę do komunikacji p2p. Jak pokazano na rysunku,



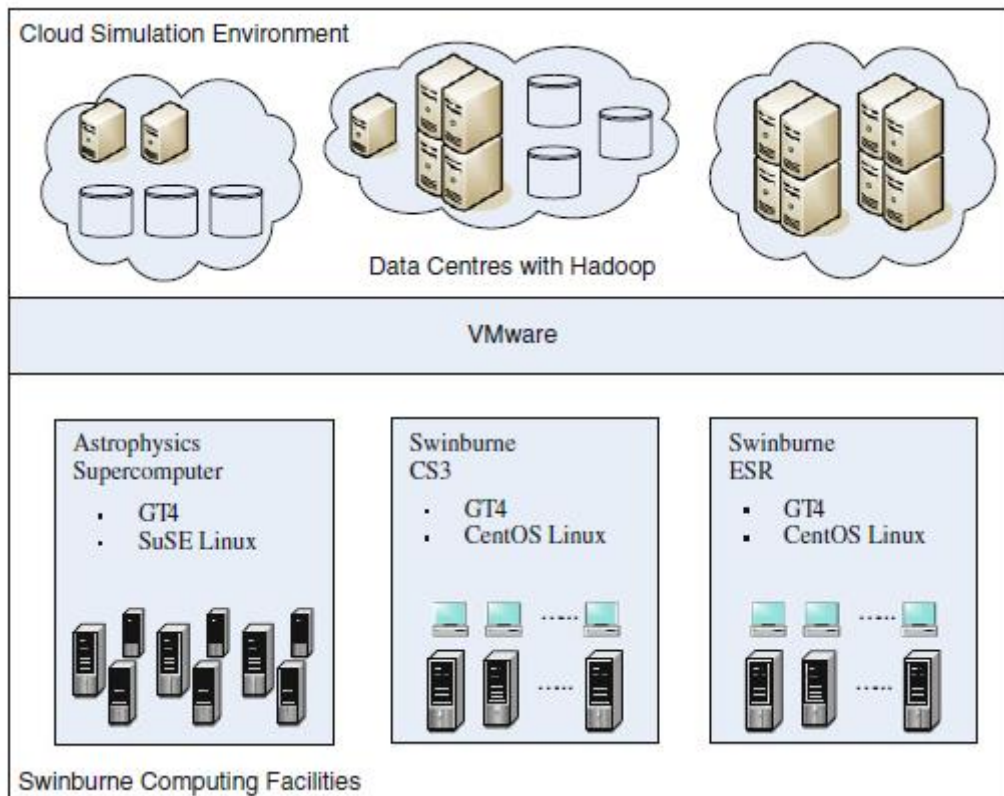
peer SwinDeW-G składa się z następujących komponentów: Komponent Zadanie zarządza zadaniami przepływu pracy. Ma dwie główne funkcje. Po pierwsze, dostarcza niezbędne informacje do Komponentu Przepływu w celu planowania i przechowuje otrzymane zadania w Repozytorium Zadań. Po drugie, określa odpowiedni czas na rozpoczęcie, wykonanie i zakończenie określonego zadania. Zasoby, których może wymagać wystąpienie zadania przepływu pracy, są przechowywane w repozytorium zasobów. Komponent Flow współdziała ze wszystkimi innymi modułami. Najpierw otrzymuje definicję przepływów pracy, a następnie tworzy definicję instancji. Po drugie, otrzymuje zadania od innych partnerów lub redystrybuuje je. Po trzecie, decyduje, czy przekazać zadanie do komponentu zadań, aby wykonać je lokalnie, czy rozestać je do innych równorzędnych użytkowników. Decyzja jest podejmowana w zależności od możliwości i obciążenia siebie i innych sąsiadów. I wreszcie, upewnia się, że wszystkie wykonania są zgodne z zależnością danych i zależnością kontroli definicji procesów, które są przechowywane w Repozytorium Procesów i Repozytorium Zadań. Menadżer Grupy jest interfejsem pomiędzy peerem a JXTA. W JXTA cała komunikacja odbywa się w ramach grupy równorzędnej, a menadżer grupy utrzymuje grupy równorzędne, do których dołączył równorzędny. Informacje o grupach równorzędnych i znajdujących się w nich równorzędnych grupach są przechowywane w repozytorium równorzędnym. Podczas gdy peer SwinDeW-G jest zaimplementowany jako usługa sieciowa, cała bezpośrednia komunikacja między peerami odbywa się za pośrednictwem p2p. Współpracownicy komunikują się w celu rozpowszechniania informacji o ich aktualnym stanie i komunikatów w celu kontroli procesu, takich jak puls, dystrybucja procesów, realizacja procesu itp. Komponent użytkownika jest interfejsem między odpowiednimi użytkownikami przepływu pracy a środowiskiem przepływu pracy. W SwinDeW-G jego podstawową funkcją jest umożliwienie użytkownikom ingerowania w instancje przepływu pracy w przypadku wystąpienia wyjątków. Globus Toolkit służy jako kontener usług sieciowych SwinDeW-G. Nie tylko sam peer SwinDeW-G jest usługą sieciową znajdującą się w Globus Toolkit, możliwości potrzebne do wykonywania niektórych zadań są również w postaci usług sieciowych, do których system ma dostęp. Oznacza to, że gdy zadanie zostanie przydzielone do peera, Globus Toolkit zostanie użyty do zapewnienia wymaganej zdolności jako usługa sieciowa dla tego zadania.

### Architektura systemu SwinDeW-C

W tej sekcji przedstawiamy architekturę systemu SwinDeW-C. SwinDeW-C (Swinburne Decentralized Workflow for Cloud) jest oparty na infrastrukturze przetwarzania w chmurze SwinCloud. SwinDeW-C dziedziczy wiele cech swojego przodka, SwinDeWG, ale ze znacznymi modyfikacjami, aby dostosować się do nowego paradygmatu przetwarzania w chmurze do zarządzania aplikacjami przepływu pracy na dużą skalę.

## Infrastruktura SwinCloud

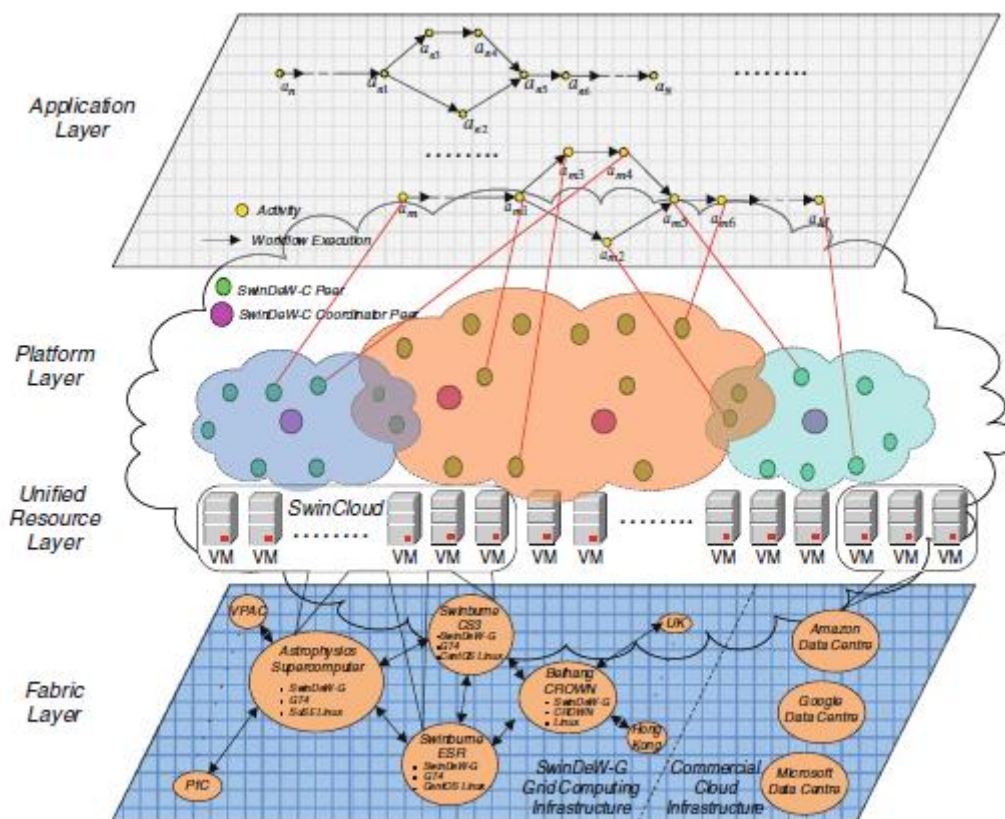
SwinCloud to środowisko symulacyjne przetwarzania w chmurze, na którym obecnie działa SwinDeW-C. Jest on zbudowany na obiektach obliczeniowych Politechniki Swinburne i wykorzystuje istniejące systemy SwinGrid. Instalujemy VMWare (VMware, 2009) na SwinGrid, aby mógł oferować zunifikowane zasoby obliczeniowe i magazynowe. Wykorzystując zunifikowane zasoby tworzymy centra danych, które mogą hostować aplikacje. W centrach danych zainstalowany jest Hadoop (2009), który może ułatwić paradygmat przetwarzania map-redukcja i zarządzanie danymi rozproszonymi. Architekturę SwinCloud przedstawiono na rysunku



## Architektura SwinDeW-C

Architekturę SwinDeW-C przedstawiono na.



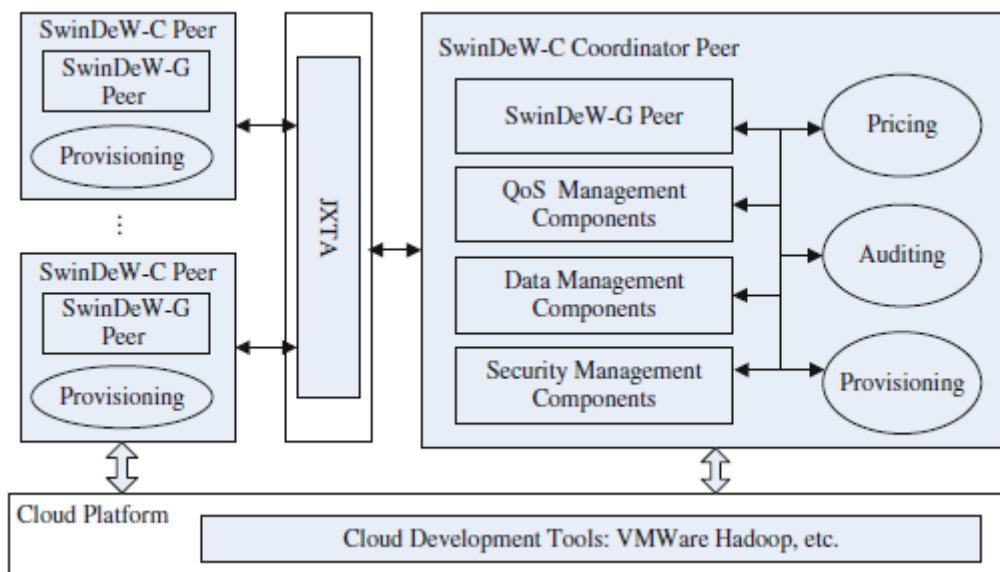


Jak omówiono wcześniej, ogólna architektura chmury obejmuje cztery podstawowe warstwy od góry do dołu: warstwę aplikacji (aplikacje użytkownika), warstwę platformy (usługi w chmurze oprogramowania pośredniczącej ułatwiającej tworzenie/wdrażanie aplikacji użytkownika), warstwę zunifikowanych zasobów (abstraktowane/enkapsulowane zasoby według wirtualizacji) i warstwy szkieletowej (fizyczne zasoby sprzętowe). W związku z tym architekturę SwinDeW-C można również odwzorować na cztery podstawowe warstwy. Tutaj przedstawiamy cykl życia abstrakcyjnej aplikacji workflow, aby zilustrować architekturę systemu. Zwróć uwagę, że tutaj skupiamy się na architekturze systemu, pominięto tutaj wprowadzenie do usług zarządzania chmurą (np. pośrednictwo, ustalanie cen, księgowość i zarządzanie maszynami wirtualnymi) oraz inne elementy funkcjonalne, które zostaną omówione w kolejnych sekcjach. Użytkownicy mogą łatwo uzyskać dostęp do portalu internetowego SwinDeW-C za pośrednictwem dowolnych urządzeń elektronicznych, takich jak komputer PC, laptop, PDA i telefon komórkowy, o ile są podłączone do Internetu. W porównaniu z SwinDeW-G, do którego można uzyskać dostęp tylko przez równorzędny SwinDeW-G z preinstalowanymi programami, portal internetowy SwinDeW-C znacznie poprawił swoją użyteczność. Na etapie budowania przepływu pracy, biorąc pod uwagę narzędzie do modelowania przepływu pracy w chmurze udostępniane przez portal internetowy w warstwie aplikacji, aplikacje przepływu pracy są modelowane przez użytkowników jako specyfikacje przepływu pracy w chmurze (składające się z definicji zadań, struktur procesów i ograniczeń QoS). Po utworzeniu specyfikacji przepływu pracy (można również udostępnić narzędzia weryfikacji statycznej, takie jak błędy struktury i ograniczenia QoS), zostaną one przesłane do dowolnego z partnerów koordynatora w warstwie platformy. Tutaj zwykły peer SwinDeW-C jest węzłem usług w chmurze, który został wyposażony w określone usługi programowe podobne do peera SwinDeW-G. Jednak podczas gdy peer SwinDeW-G jest wdrażany na samodzielnej maszynie fizycznej ze stałymi jednostkami obliczeniowymi i przestrzenią pamięci, peer SwinDeW-C jest wdrażany na maszynie wirtualnej, której moc obliczeniowa może być dynamicznie skalowana zgodnie z żądaniem

zadania. Jeśli chodzi o peery koordynatora SwinDeW-C, są to super węzły wyposażone w dodatkowe usługi zarządzania przepływem pracy w porównaniu ze zwykłymi peerami SwinDeW-C. Szczegóły dotyczące równorzędnych partnerów SwinDeW-C zostaną przedstawione w następnej sekcji. Na etapie tworzenia instancji w czasie wykonywania specyfikację przepływu pracy w chmurze można przesłać do dowolnego partnera koordynującego SwinDeW-C. Następnie zadania związane z przepływem pracy zostaną przydzielone odpowiednim partnerom poprzez komunikację typu peer-to-peer między rówieśnikami SwinDeW-C. Ponieważ zarządzanie peerami, takie jak peer join, peer leave i peer search, a także mechanizm wykonywania przepływu pracy oparty na p2p, jest taki sam jak w środowisku systemu SwinDeW-G. W związku z tym pominięto tutaj szczegółowe wprowadzenie, które można znaleźć w (Yang i in., 2007). Przed wykonaniem przepływu pracy, partner koordynatora przeprowadzi proces oceny przesłanych instancji przepływu pracy w chmurze, aby określić, czy można je zaakceptować, czy nie określone niefunkcjonalne wymagania QoS w obecnym modelu cenowym. Ogólnie przyjmuje się, że wymagania funkcjonalne mogą być zawsze spełnione przy teoretycznie nieograniczonej skalowalności chmury. W przypadku, gdy użytkownicy muszą uruchomić własne specjalne programy, mogą je przesłać za pośrednictwem portalu internetowego, a programy te zostaną automatycznie wdrożone w centrum danych przez menedżera zasobów. W tym przypadku proces negocjacji pomiędzy użytkownikiem a chmurowym systemem workflow może być przeprowadzony, jeśli przesłana przez użytkownika instancja workflow nie jest akceptowalna dla systemu workflow ze względu na niedopuszczalną ofertę budżetów lub terminów. Ostatecznym wynikiem negocjacji będą skompromitowane wymagania QoS lub nieudane przesłanie wystąpienia przepływu pracy w chmurze. Jeśli wszystkie instancje zadań zostały pomyślnie przydzielone (tzn. komunikaty akceptacyjne są odsyłane do koordynującego peera ze wszystkich przydzielonych peerów), instancja workflow w chmurze może zostać ukończona ze spełnieniem zarówno funkcjonalnych, jak i niefunkcjonalnych wymagań QoS (jeśli bez wyjątków). W związku z tym instancja przepływu pracy w chmurze jest pomyślnie tworzona. Wreszcie, na etapie wykonywania w czasie wykonywania, każde zadanie jest wykonywane przez peera SwinDeW-C. W przypadku przetwarzania w chmurze podstawowe heterogeniczne zasoby są zwirtualizowane jako zasoby zunifikowane (maszyny wirtualne). Każdy peer wykorzystuje moc obliczeniową zapewnianą przez jego maszynę wirtualną, którą można łatwo skalować zgodnie z wymaganiami zadań przepływu pracy. Jak widać na zunifikowanej warstwie zasobów na rysunku powyżej, SwinCloud jest zbudowany na poprzedniej infrastrukturze SwinGrid w warstwie Fabric. Tymczasem niektóre maszyny wirtualne mogą być tworzone z zewnętrznymi komercyjnymi dostawcami usług chmurowych IaaS (infrastruktura jako usługa), takimi jak Amazon, Google i Microsoft. Podczas wykonywania przepływu pracy w chmurze zadania zarządzania przepływem pracy, takie jak zarządzanie QoS, zarządzanie danymi i zarządzanie bezpieczeństwem, są wykonywane przez partnerów koordynatora w celu osiągnięcia satysfakcjonującej wydajności systemu. Użytkownicy mogą w dowolnym momencie uzyskać dostęp do ostatecznych wyników, a także bieżących informacji o przesłanych instancjach przepływu pracy za pośrednictwem portalu internetowego SwinDeW-C.

### **Architektura peerów SwinDeW-C**

W tej sekcji przedstawimy architekturę peera SwinDeW-C. Jak opisaliśmy powyżej, SwinDeW-C jest rozwijany w oparciu o SwinDeW-G, gdzie peer SwinDeW-C odziedziczył większość komponentów peera SwinDeW-G, w tym elementy zarządzania zadaniami, zarządzanie przepływem, repozytoria i zarządzanie grupami. W związku z tym peer SwinDeW-G odgrywa rolę rdzenia peera SwinDeW-C, który zapewnia podstawowe elementy zarządzania przepływem pracy i elementy komunikacji między rówieśnikami. Jednak pewne ulepszenia są również wprowadzane na peerach SwinDeW-C, aby dostosować je do środowiska przetwarzania w chmurze. Architekturę sąsiadów SwinDeW-C przedstawiono na rysunku



Po pierwsze, w odróżnieniu od peera SwinDeW-G, peer SwinDeW-C działa na platformie chmury. Platforma chmurowa składa się z ujednoliconych zasobów, co oznacza, że możliwości obliczeniowe i przechowywania danych peera SwinDeW-C mogą być dynamicznie skalowane w górę lub w dół w zależności od wymagań aplikacji. Ujednolicone zasoby są oferowane przez dostawców usług w chmurze i zarządzane w pulach zasobów, dlatego każdy peer SwinDeW-C ma komponent do udostępniania, który umożliwia dynamiczne aplikowanie i zwalnianie zasobów w chmurze. Po drugie, w środowisku chmury obliczeniowej różni dostawcy usług w chmurze mogą mieć inny model kosztów, dlatego musimy ustawić peera koordynatora w ramach każdego dostawcy usług w chmurze. Element równorzędny koordynatora ma komponenty wyceny i audytu, które mogą koordynować udostępnianie zasobów wszystkich elementów równorzędnych znajdujących się w tym dostawcy usług. Wreszcie, koordynator SwinDeW-C ma również nowe komponenty funkcjonalne związane z zarządzaniem przepływem pracy w chmurze. Jak przedstawiono w rozdziale 13.2.2, system ma nowe wymagania dotyczące obsługi aplikacji przepływu pracy na dużą skalę. Aby spełnić te nowe wymagania, do węzła koordynatora SwinDeW-C dodano komponenty zarządzania QoS, zarządzania danymi i zarządzania bezpieczeństwem.

### Nowe komponenty w SwinDeW-C

W tej sekcji przedstawiamy nowe komponenty w SwinDeW-C. Jako trzy wymagania systemowe, wprowadzono trzy nowe komponenty funkcjonalne, w tym zarządzanie QoS, zarządzanie danymi i zarządzanie bezpieczeństwem.

### Zarządzanie QoS w SwinDeW-C

Podstawowy wymóg zapewnienia zadowalającej tymczasowej QoS (a także innych ograniczeń QoS) obejmuje trzy podstawowe zadania: ustawianie ograniczeń QoS, monitorowanie wykonywania przepływu pracy pod kątem naruszeń ograniczeń QoS oraz obsługę naruszeń ograniczeń QoS. Weźmy na przykład czasowe ograniczenia QoS, wprowadzono nowy komponent zarządzania QoS w SwinDeW-C. Temporal Constraint Setting: W komponencie zarządzania QoS SwinDeW-C strategia probabilistyczna jest przeznaczona do ustawiania tymczasowych ograniczeń QoS w czasie budowania przepływu pracy). W szczególności, w przypadku modelu spójności czasowej opartego na prawdopodobieństwie, jedno globalne lub kilka gruboziarnistych ograniczeń czasowych jest przypisywanych na podstawie wyniku negocjacji między klientami a dostawcami usług. Następnie

drobnoziarniste ograniczenia czasowe dla jednostki działania przepływu pracy mogą być wyprowadzane automatycznie na podstawie tych gruboziarnistych. Wybór punktu kontrolnego i weryfikacja czasowa: W czasie wykonywania przepływu pracy dostępna jest strategia wyboru punktu kontrolnego i strategia weryfikacji czasowej w celu monitorowania wykonania przepływu pracy pod kątem naruszenia ograniczeń czasowych. Weryfikacja czasowa ma na celu sprawdzenie poprawności czasowej stanów wykonania przepływu pracy (wykrywanie naruszeń czasowych) przy założonym modelu spójności czasowej. Tymczasem, w celu zaoszczędzenia ogólnych kosztów zarządzania QoS, weryfikację czasową należy przeprowadzać tylko na wybranych punktach aktywności. W SwinDeW-C zastosowano strategię wyboru punktów kontrolnych opartą na minimalnej redundancji czasowej, która pozwala wybrać tylko niezbędne i wystarczające punkty kontrolne (te, w których dochodzi do czasowych uchybień). Obsługa wyjątków: po wykryciu tymczasowego naruszenia wymagane są strategie obsługi wyjątków w celu odzyskania stanów błędów. W przeciwieństwie do błędów funkcjonalnych, którym zwykle zapobiegają zduplikowane instancje lub które są obsługiwane przez wycofanie i ponowne wykonanie, niefunkcjonalne błędy QoS, takie jak naruszenia czasowe, można naprawić tylko poprzez kompensację, tj. w celu zmniejszenia lub idealnego usunięcia bieżących opóźnień czasowych poprzez skrócenie czasu trwania kolejne czynności workflow. Ponieważ poprzednie czynności zostały już zakończone, w realnym świecie nie ma możliwości skrócenia czasu ich wykonywania. W SwinDeW-C, w przypadku drobnych naruszeń czasowych, stosowana jest strategia TDA (alokacja deficytu czasu) (Chen & Yang, 2007a), która może usunąć bieżące deficyty czasu poprzez pożyczanie nadmiarowości czasu na kolejne czynności. Jeśli chodzi o poważne naruszenia czasowe, stosuje się strategię ACOWR (optymalizację kolonii mrówek w oparciu o dwuetapowe lokalne zmiany harmonogramu przepływu pracy), która może skrócić czas trwania kolejnych segmentów przepływu pracy poprzez zmianę harmonogramu przepływu pracy w oparciu o optymalizację kolonii mrówek. W SwinDeW-C, dzięki stałemu monitorowaniu instancji przepływu pracy i skutecznej obsłudze naruszeń czasowych podczas wykonywania przepływu pracy, można zapewnić zadowalającą czasową jakość QoS przy niskich wskaźnikach naruszeń zarówno globalnych, jak i lokalnych ograniczeń czasowych. Podobnie jak w przypadku tymczasowego zarządzania QoS, badane są zadania zarządzania dla innych ograniczeń QoS. Tymczasem, ponieważ niektóre z nich, takie jak koszty i bezpieczeństwo, są częściowo uwzględnione w komponentach zarządzania danymi i zarządzania bezpieczeństwem, niektóre funkcje będą wspólne dla tych komponentów.

### **Zarządzanie danymi w SwinDeW-C**

Komponent zarządzania danymi w SwinDeW-C składa się z trzech podstawowych zadań: przechowywania danych, umieszczania danych i replikacji danych. Przechowywanie danych: W tym komponentcie usprawniono opłacalną strategię przechowywania danych opartą na zależnościach w celu przechowywania danych aplikacji. Strategia wykorzystuje informacje o pochodzeniu danych instancji przepływu pracy. Pochodzenie danych w przepływach pracy jest rodzajem ważnych metadanych, w których rejestrowane są zależności między zbiorami danych. Zależność przedstawia pochodną relację między zestawami danych aplikacji. W systemach przepływu pracy w chmurze, po wykonaniu zadań, niektóre pośrednie zestawy danych mogą zostać usunięte, aby zaoszczędzić na kosztach przechowywania, ale czasami trzeba je ponownie wygenerować w celu ponownego użycia lub ponownej analizy. Pochodzenie danych rejestruje informacje o tym, jak wygenerowano zestawy danych. Ponadto regeneracja pośrednich zestawów danych z danych wejściowych może być bardzo czasochłonna, a zatem wiązać się z wysokimi kosztami obliczeń. W przypadku informacji o pochodzeniu danych regeneracja wymagającego zestawu danych może zamiast tego rozpocząć się od niektórych przechowywanych pośrednich zestawów danych. W systemie przepływu pracy w chmurze pochodzenie danych jest rejestrowane podczas wykonywania przepływu pracy. Wykorzystując proveniencję danych, możemy zbudować Intermediate Data Dependency Graph (IDG) w oparciu o

proweniencję danych. Wszystkie pośrednie zbiory danych raz wygenerowane w systemie, przechowywane lub usunięte, ich odniesienia są rejestrowane w IDG. Na podstawie IDG możemy obliczyć koszt generowania każdego zestawu danych w przepływach pracy w chmurze. Porównując koszt generowania i koszt pamięci masowej, strategia pamięci masowej może automatycznie zdecydować, czy zestaw danych powinny być przechowywane lub usuwane w systemie w chmurze w celu zmniejszenia kosztów systemu, bez względu na to, czy ten zbiór danych jest nowym zbiorem danych, zregenerowanym zbiorem danych czy zbiorem danych przechowywanym w systemie.

Rozmieszczenie danych: W tym komponencie strategia rozmieszczania danych jest ułatwiona w celu umieszczenia danych aplikacji, które mogą zmniejszyć ruch danych podczas wykonywania przepływów pracy. W systemach cloud computing infrastruktura jest ukryta przed użytkownikami. Dlatego w przypadku danych aplikacji system zdecyduje, gdzie je przechowywać. W strategii wstępnie dostosowujemy algorytm grupowania k-średnich do umieszczania danych w systemach workflow w chmurze w oparciu o zależność danych. Przepływy pracy w chmurze mogą być złożone, jedno zadanie może wymagać wielu zestawów danych do wykonania; ponadto jeden zestaw danych może być również wymagany przez wiele zadań. Jeśli niektóre zestawy danych są zawsze używane razem przez wiele zadań, mówimy, że te zestawy danych są od siebie zależne. W naszej strategii staramy się przechowywać te zestawy danych w jednym centrum danych, tak aby podczas planowania zadań w tym centrum danych większość, jeśli nie wszystkie potrzebne dane, była przechowywana lokalnie. Nasza strategia rozmieszczania danych obejmuje dwa algorytmy, jeden dla etapu kompilacji, a drugi dla etapu wykonywania naukowych przepływów pracy. W algorytmie etapu budowania budujemy macierz zależności dla wszystkich danych aplikacji, która reprezentuje zależności między wszystkimi zestawami danych. Następnie używamy algorytmu BEA do klastrowania macierzy i partycjonowania tak, aby zestawy danych w każdej partycji były od siebie wysoce zależne. Rozdzielamy partycje na k centrów danych, które początkowo są partycjami algorytmu k-średnich na etapie czasu wykonywania. W czasie wykonywania nasz algorytm klastrowania zajmuje się nowo wygenerowanymi danymi, które będą potrzebne innym zadaniom. Dla każdego nowo wygenerowanego zestawu danych obliczamy jego zależności ze wszystkimi k centrów danych i przenosimy dane do centrum danych, które ma z nim największą zależność. Replikacja danych: W tym komponencie usprawniono dynamiczną strategię replikacji danych, aby zagwarantować bezpieczeństwo danych i szybki dostęp do danych w systemach przepływu pracy w chmurze. Przechowywanie niektórych replik danych aplikacji ma kluczowe znaczenie dla bezpieczeństwa danych w pamięci masowej w chmurze. Replikacja statyczna może zagwarantować niezawodność da poprzez utrzymywanie stałej liczby replik danych aplikacji, ale w środowisku chmury różne dane aplikacji mają różne współczynniki wykorzystania, gdzie strategia musi być dynamiczna, aby replikować dane aplikacji na podstawie ich użycia stawki. W aplikacjach przepływu pracy na dużą skalę wiele równoległych zadań będzie jednocześnie uzyskiwać dostęp do tego samego zestawu danych w jednym centrum danych. Ograniczenie mocy obliczeniowej i przepustowości w tym centrum danych byłoby wąskim gardłem dla całego systemu workflow w chmurze. Jeśli mamy kilka replik w różnych centrach danych, to wąskie gardło zostanie wyeliminowane. W związku z tym replikacja danych zawsze będzie przechowywać stałą liczbę kopii wszystkich zestawów danych w różnych centrach danych, aby zagwarantować niezawodność i dynamicznie dodawać nowe repliki dla każdego zestawu danych, aby zagwarantować dostępność danych. Co więcej, rozmieszczenie replik jest oparte na zależności danych, która jest taka sama, jak w przypadku komponentu rozmieszczania danych, a liczba replik, które powinien mieć zestaw danych, zależy od wskaźnika wykorzystania tego zestawu danych.

### **Zarządzanie bezpieczeństwem w SwinDeW-C**

Aby rozwiązać problemy bezpieczeństwa dla bezpiecznego działania SwinDeW-C, zaprojektowano komponent zarządzania bezpieczeństwem. Jako typ typowego rozproszonego systemu

obliczeniowego, zarządzanie zaufaniem dla peerów SwinDeW-C jest bardzo ważne i odgrywa najważniejszą rolę w zarządzaniu bezpieczeństwem. Poza tym istnieje kilka innych kwestii bezpieczeństwa, które powinniśmy rozważyć z perspektywy użytkownika i danych. W szczególności w komponencie zarządzania bezpieczeństwem istnieją trzy moduły: zarządzanie zaufaniem, zarządzanie użytkownikami i system zarządzania szyfrowaniem. Zarządzanie zaufaniem: Celem modułu zarządzania zaufaniem jest zarządzanie relacjami między jednym peerem SwinDeW-C a sąsiednimi peerami. Na przykład, aby przetworzyć instancję przepływu pracy, peer SwinDeW-C musi współpracować z sąsiednimi peerami, aby uruchomić tę instancję. Ze względu na wysoki poziom wymagań QoS dużych skalowanych aplikacji workflow, równorzędne zarządzanie w SwinDeW-C powinno być wspierane przez zarządzanie zaufaniem podczas pracy workflow. Moduł zarządzania zaufaniem działa jak konsultant. Ten moduł może oceniać niektóre zadania i udzielać porad dotyczących współpracy między jednym a innymi rówieśnikami dla każdego wystąpienia określonego zadania. Po pierwsze, ocena rówieśnicza dokonuje oceny zaufania innych sąsiednich rówieśników. Po drugie, ewaluacja zadania dokonuje oceny ponownego przydziału zadania innym rówieśnikom. Następnie obie oceny zostaną połączone z oceną zaufania, aby dojść do wniosku, czy ten sąsiadujący partner ma wystarczające zaufanie, aby podjąć to zadanie. Poza tym projektujemy bazę reguł. Na przykład określone zadanie nie może być przypisane do jednego konkretnego sąsiada z sąsiedztwa i jest to prosta zasada. Baza reguł jest uzupełnieniem poprzedniej oceny zaufania opartej na wartościach, aby dopasować ją do rzeczywistej sytuacji. Zarządzanie użytkownikami: moduł zarządzania użytkownikami jest niezbędnym elementem każdego systemu. W SwinDeW-C baza użytkowników to baza danych, która przechowuje wszystkie informacje o tożsamości użytkowników i logach, które przesyłają żądania usług. Ponadto menedżer uprawnień kontroluje uprawnienia użytkowników do przesyłania niektórych specjalnych żądań usług. System zarządzania szyfrowaniem: Biorąc pod uwagę, że sąsiedzi SwinDeW-C znajdują się w różnych geograficznych sieciach lokalnych, ważne jest zapewnienie bezpieczeństwa danych w procesie przesyłania danych poprzez szyfrowanie. W SwinDeW-C wybieramy narzędzie PGP GnuPG (<http://www.gnupg.org>), aby zapewnić bezpieczną komutację. Podsumowując, poza powyższymi trzema nowymi elementami funkcjonalnymi, SwinDeW-C zawiera również wspólne elementy funkcjonalne chmury, takie jak pośrednictwo, ustalanie cen, audyt i zarządzanie maszynami wirtualnymi. Szczegółowy opis można znaleźć w i dlatego został pominięty w tym artykule.

### **Prototyp systemu SwinDeW-C**

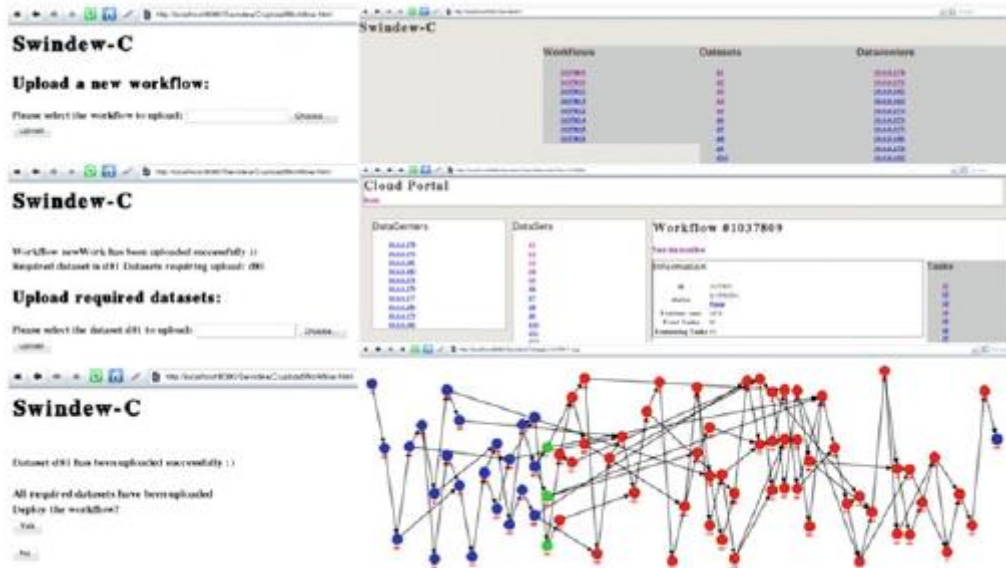
Na podstawie omówionego powyżej projektu zbudowaliśmy prymitywny prototyp SwinDeW-C. Prototyp został opracowany w Javie i obecnie działa w środowisku symulacyjnym SwinCloud. W prototypie SwinDeW-C odziedziczyliśmy większość funkcji SwinDeW-G, a następnie wdrożyliśmy nowe komponenty SwinDeW-C, aby mógł on dostosować się do środowiska chmury obliczeniowej. Ponadto zbudowaliśmy portal WWW dla SwinDeW-C, dzięki któremu użytkownicy i administrator systemu mogą uzyskać dostęp do zasobów chmury i zarządzać aplikacjami SwinDeW-C. Portal internetowy udostępnia wiele interfejsów, które wspierają zarówno użytkowników systemu, jak i administratorów w następujących zadaniach, szczególnie dla użytkownika systemu:

- a) przeglądać istniejące zbiory danych znajdujące się w centrach danych różnych dostawców usług w chmurze;
- (b) przysyłać swoje dane aplikacji do i pobierać dane wynikowe z pamięci w chmurze;
- (c) tworzenie i wdrażanie przepływów pracy w systemie SwinDeW-C za pomocą narzędzi do modelowania;
- d) monitorować wykonanie przepływów pracy.

Do administrowania systemem:

- a) koordynować wykonywanie przepływów pracy poprzez uruchamianie strategii planowania;
- b) zarządzać zbiorami danych aplikacji poprzez uruchamianie strategii umieszczania danych;
- (c) obsłużyć wyjątki wykonania, uruchamiając strategię dostosowania przepływu pracy.

Niektóre interfejsy portalu internetowego pokazano poniżej.



## Powiązana praca

Ponieważ badania nad systemami zarządzania przepływem pracy w chmurze są na początkowym etapie, trudno jest obecnie przeprowadzić bezpośrednie porównanie SwinDeW-C z innymi. Większość obecnych projektów dotyczy albo ogólnego wdrażania przetwarzania w chmurze, albo skupia się na pewnych konkretnych aspektach, takich jak zarządzanie danymi w chmurze. Istnieją pewne badania dotyczące aplikacji w chmurze intensywnie korzystających z danych, takich jak wczesne doświadczenia, takie jak Nimbus i Cumulus. W porównaniu z rozproszonymi systemami obliczeniowymi, takimi jak klastery i siatki, system przetwarzania w chmurze przynosi korzyści kosztowe, które pokazują, że przetwarzanie w chmurze może zwiększyć pojemność klastrów z korzyścią kosztową. Wykorzystując model kosztów chmury Amazon i oprogramowanie pośredniczące BOINC do obliczeń wolontariuszy, prace w analizują korzyści kosztowe przetwarzania w chmurze w porównaniu z przetwarzaniem siatkowym. Jeśli chodzi o korzyści kosztowe, prace Deelmana, Singha, Livny, Berrimana i Gooda (2008) pokazują, że przetwarzanie w chmurze oferuje opłacalne rozwiązanie dla aplikacji intensywnie korzystających z danych, takich jak naukowe przepływy pracy. Praca bada wykorzystanie chmury obliczeniowej do naukowych przepływów pracy, koncentrując się na szeroko stosowanej aplikacji astronomicznej - Montage. Projekt Cloudbus (<http://www.gridbus.org/cloudbus/>) prowadzony w laboratorium CLOUDS na Uniwersytecie w Melbourne pracuje nad nową, uogólnioną i rozszerzalną platformą symulacji chmury o nazwie CloudSim, która może umożliwić bezproblemowe modelowanie, symulację i eksperymentowanie z infrastrukturami przetwarzania w chmurze i usługami zarządzania.

Biorąc pod uwagę istniejące projekty dotyczące wielu systemów przepływu pracy w sieciach opracowanych w ostatnich latach, wielu badaczy i praktyków zgadza się, że systemy przepływu pracy w chmurze mogą być budowane na środowiskach gridowych, a nie od zera. Na przykład zestaw

narzędzi CloudSim wykorzystany w projekcie Cloudbus jest wdrażany poprzez programowe rozszerzenie podstawowych funkcjonalności udostępnianych przez GridSim używany w projekcie Gridbus (2010). Dlatego w tym rozdziale dokonamy przeglądu niektórych reprezentatywnych systemów przepływu pracy w sieci i skupimy się na powiązanych funkcjach omówionych w tym dokumencie, takich jak architektura planowania przepływu pracy, QoS, zarządzanie danymi i bezpieczeństwem. W szczególności badamy Gridbus (2010), Pegasus (2010), Taverna (2010), GraDS (2010), ASKALON (2010), GridAnt (2010), Triana (2010), GridFlow (2010) i Kepler (2010). W przypadku architektury planowania przepływu pracy Pegasus, Taverna, GraDS i Kepler używają architektury scentralizowanej; Gridbus i GridFlow używają architektury hierarchicznej; ASKALON i Triana używają zdecentralizowanej architektury. Uważa się, że schematy scentralizowane tworzą bardziej wydajne harmonogramy, a schematy zdecentralizowane mają lepszą skalowalność, podczas gdy schematy hierarchiczne są ich kompromisami. Podobnie jak SwinDeW-G, SwinDeW-C wykorzystuje ustrukturyzowany zdecentralizowany schemat planowania przepływu pracy. SwinDeW-G ma na celu wykorzystanie strategii opartej na wydajności w celu osiągnięcia ogólnego zrównoważenia obciążenia całego systemu poprzez dystrybucję zadań do najmniej obciążonych sąsiadów. Jeśli chodzi o ograniczenia QoS (jakość usług), większość przepływów pracy w sieci nie obsługuje tej funkcji. Gridbus obsługuje ograniczenia QoS, w tym terminy realizacji zadań i minimalizację kosztów, GraDS i GridFlow wykorzystują głównie szacowany czas wykonania aplikacji, a ASKALON obsługuje ograniczenia i właściwości określone przez użytkowników lub predefiniowane. Obecnie SwinDeW-C obsługuje ograniczenia QoS w oparciu o terminy realizacji zadań. Jeśli chodzi o odporność na awarie, na poziomie zadań Gridbus, Taverna, ASKALON, Karajan, GridFlow i Kepler korzystają z alternatywnych zasobów; Taverna, ASKALON i Karajan ponawiają próbę; Na poziomie workflow ratowniczy workflow jest używany przez ASKALON i Kepler; Obsługa wyjątków zdefiniowana przez użytkownika jest używana przez Karajan i Kepler. Pegasus, GridAnt i Triana stosują odpowiednio swoje konkretne strategie. Dla porównania, SwinDeW-C wykorzystuje skuteczną weryfikację ograniczeń czasowych do wykrywania i postępowanie z naruszeniami czasowymi. Jeśli chodzi o zarządzanie danymi, Kepler ma własną, zorientowaną na aktora metodę modelowania danych, która jest przeznaczona dla dużych danych w środowisku gridowym. Ma dwóch aktorów Grid, odpowiednio nazwanych FileFetcher i FileStager. Aktorzy ci używają GridFTP do pobierania plików z lub przenoszenia plików do zdalnych lokalizacji w sieci Grid. Pegasus opracował kilka algorytmów umieszczania danych w środowisku grid i wykorzystuje system RLS (Replica Location Service) do zarządzania danymi w czasie wykonywania. W Pegasusie dane są asynchronicznie przenoszone do zadań na żądanie, aby skrócić czas oczekiwania na wykonanie i dynamicznie usuwać dane, których zadanie już nie potrzebuje, aby zmniejszyć wykorzystanie pamięci. W Gridbus system przepływu pracy ma kilka algorytmów planowania dla aplikacji intensywnie korzystających z danych w środowisku gridowym w oparciu o Grid Resource Broker. Algorytmy są zaprojektowane w oparciu o różne teorie (GA, MDP, SCP, Heurystyka), aby dostosować się do różnych przypadków użycia. Taverna zaproponowała nowy język definicji procesów, Sculf, który może modelować dane aplikacji w przepływie danych. Traktuje przepływ pracy jako wykres procesorów, z których każdy przenosi zestaw danych wejściowych do zestawu danych wyjściowych. ASKALON to system workflow przeznaczony do planowania. Łączy narzuty obliczeniowe i nakłady związane z transferem danych, aby uzyskać „wagę” wartości. Nie dyskryminuje zasobu obliczeniowego i hosta danych. ASKALON posiada również własny język definicji procesów o nazwie AGWL. Triana to system przepływu pracy oparty na środowisku rozwiązywania problemów, które umożliwia aplikację naukową wymagającą dużej ilości danych do wykonania. W przypadku sieci grid posiada ona niezależną warstwę pośredniczącą abstrakcji, zwaną Grid Application Prototype (GAP), umożliwiającą użytkownikom reklamowanie, odkrywanie i komunikowanie się z usługami sieciowymi i peer-to-peer (p2p). Triana używa również RLS do zarządzania danymi w czasie wykonywania. GridFlow to system przepływu pracy, który wykorzystuje system oparty na agentach do zarządzania zasobami gridu. Traktuje transfer



danych do zasobów obliczeniowych i archiwizację do zasobów pamięci jako rodzaje zadań przepływu pracy. Jeśli chodzi o zarządzanie bezpieczeństwem, Globus wykorzystuje kryptografię klucza publicznego (znaną również jako kryptografię asymetryczną) jako podstawę zarządzania bezpieczeństwem, która reprezentuje główny nurt w obszarze bezpieczeństwa sieci. Globus używa certyfikatów zakodowanych w formacie certyfikatu X.509, ustalonym standardowym formacie danych. Certyfikaty te mogą być współużytkowane przez oprogramowanie oparte na kluczu publicznym, w tym komercyjne przeglądarki internetowe firm Microsoft i Netscape. Międzynarodowa Federacja Zaufania Grid (IGTF) (<http://www.igtf.net/>) jest zewnętrzną dostawcą usług zaufania do sieci grid, który ma na celu ustanowienie wspólnych zasad i wytycznych między członkami swoich organów zarządzających polityką (PMA). IGTF nie zapewnia asercji tożsamości, ale zapewnia, że w zakresie statutu IGTF, asercje wydane przez akredytowane organy któregośkolwiek z jej członków PMA mogą spełniać lub przekraczać profil uwierzytelniania właściwy dla akredytowanego organu. Europejski projekt GridTrust (<http://www.gridtrust.eu/gridtrust/>) to nowatorski i ambitny projekt, który zapewnia nowe usługi bezpieczeństwa w warstwie oprogramowania pośredniego GRID. GridTrust opracowuje usługę kontroli użytkownika, aby monitorować wykorzystanie zasobów w dynamicznych organizacjach wirtualnych (VO), egzekwować zasady użytkownika w czasie wykonywania i zgłaszać naruszenia zasad kontroli użytkownika. Ta usługa zapewnia dynamiczną kontrolę wykorzystania bezpieczeństwa sieci w tradycyjnych, sztywnych modelach autoryzacji. Inne usługi struktury bezpieczeństwa obejmują edytor Grid Security Requirements, który umożliwia właścicielom i użytkownikom VO definiowanie zasad bezpieczeństwa; usługę Secure-Aware Resource Broker, która pomaga tworzyć VO w oparciu o usługi z kompatybilnymi zasadami bezpieczeństwa; oraz zaawansowaną Usługę Menedżera Reputacji do rejestrowania przeszłych zachowań właścicieli i użytkowników VO jako danych uwierzytelniających reputację.

## **Wnioski**

Zaawansowane aplikacje przepływu pracy na dużą skalę są powszechnie spotykane zarówno w obszarach e-biznesu, jak i e-nauki. Systemy przepływu pracy zbudowane na wysokowydajnych infrastrukturach obliczeniowych, takich jak klastry, p2p i przetwarzanie siatkowe, są często stosowane do wspomagania automatyzacji procesów w aplikacjach przepływu pracy na dużą skalę. Jednak dwa podstawowe wymagania, w tym skalowalne zasoby i zdecentralizowane zarządzanie, nie zostały jak dotąd dobrze rozwiązane. Ostatnio, wraz z pojawieniem się chmury obliczeniowej, która jest nowym paradygmatem obliczeniowym, który może zapewnić praktycznie nieograniczone zasoby obliczeniowe o łatwej skali, system przepływu pracy w chmurze jest obiecującym nowym rozwiązaniem i dlatego zasługuje na systematyczne badanie. W tym rozdziale zaprezentowany został SwinDeW-C, nowatorski system przepływu pracy w chmurze oparty na sieci peer-to-peer. SwinDeW-C nie jest zbudowany od zera, ale na swoim poprzedniku SwinDeW-G (system przepływu pracy oparty na sieci p2p). Aby dostosować się do paradygmatu przetwarzania w chmurze i ułatwić zarządzanie aplikacjami workflow na dużą skalę, w poprzednim systemie SwinDeW-G wprowadzono istotne modyfikacje. W szczególności oryginalna warstwa tkaniny SwinDeW-G jest dziedziczona wraz z rozszerzeniem zewnętrznych komercyjnych dostawców usług w chmurze. Tymczasem zasoby bazowe są zwirtualizowane w ujednoliconej warstwie zasobów; elementy funkcjonalne, w tym zarządzanie QoS, zarządzanie danymi i zarządzanie bezpieczeństwem są dodawane lub ulepszone w warstwie platformy, aby wspierać zarządzanie aplikacjami przepływu pracy na dużą skalę; interfejs użytkownika został zmodyfikowany w celu obsługi dostępu do Internetu (przeglądarki internetowej). W tym rozdziale opisano architekturę systemu SwinDeW-C i jego nowe funkcje zarządzania instancjami i aplikacjami przepływu pracy intensywnie korzystającymi z danych/obliczeń. Zademonstrowano prototypowy system SwinDeW-C, ale wciąż go rozwijamy. W przyszłości więcej funkcjonalnych komponentów zostanie zaprojektowanych i wdrożonych w celu zwiększenia możliwości SwinDeW-C. W międzyczasie

przeprowadzone zostanie również porównanie między SwinDeW-C a innymi systemami przepływu pracy w oparciu o statystyki pomiarów wydajności, takich jak wskaźnik sukcesu, wskaźnik tymczasowych naruszeń, przepustowość systemu i inne.