

## **Rola technologii Grid Computing w Cloud Computing**

Dziedziny Grid, Utility i Cloud Computing mają zestaw wspólnych celów w zakresie wykorzystywania współdzielonych zasobów, aby optymalnie sprostać różnorodnym wymaganiom w sposób ekonomiczny i terminowy. Ponieważ Grid Computing rozpoczął swoją technologiczną podróż około dekadę wcześniej niż Cloud Computing, Chmura może korzystać z technologii i doświadczenia Grid w budowaniu infrastruktury dla przetwarzania rozproszonego. Nasze porównanie Grid i Cloud zaczyna się od ich podstawowych cech i modeli interakcji z klientami, konsumentami zasobów i dostawcami. Następnie badane są podobieństwa i różnice w warstwach architektonicznych oraz kluczowe wzorce użytkowania. Następnie dogłębnie przyjrzymy się technologiom i najlepszym praktykom, które mają zastosowanie od sieci grid po przetwarzanie w chmurze, w tym planowanie, orientację na usługi, bezpieczeństwo, zarządzanie danymi, monitorowanie, interoperacyjność, symulację i wsparcie autonomiczne. Na koniec oferujemy wgląd w to, w jaki sposób te techniki pomogą rozwiązać obecne wyzwania, przed którymi stoi przetwarzanie w chmurze.

### **Wstęp**

Przetwarzanie w chmurze wykorzystuje postępy w sprzęcie komputerowym i modelach programowania, które można połączyć, aby zapewnić rozwiązania użytkowe problemów obliczeniowych na dużą skalę. Na poziomie sprzętowym ostatnie pół wieku przyniosło obfity postęp w zakresie mocy obliczeniowej. Wynika to z wielu ulepszeń na poziomie procesora, a w ostatnich latach z dostępności tanich układów wielordzeniowych. Dodatkowe postępy w szybkich połączeniach międzysieciowych o niskich opóźnieniach umożliwiły tworzenie lokalnych klastrów na dużą skalę do przetwarzania rozproszonego oraz rozszerzenie na rozległe klastry współpracujące w sieci Grid. Obecnie niedawna dostępność wsparcia sprzętowego do wirtualizacji platformy na maszynach standardowych stanowi kluczowy element umożliwiający przetwarzanie w chmurze. Modele oprogramowania poruszają się w sposób lockstep, aby dorównać postępom w sprzęcie. Istnieje duże doświadczenie praktyczne we wdrażaniu rozwiązań przetwarzania rozproszonego oraz we wspieraniu modeli programowania równoległego w klastrach. Modele te działają teraz w celu wykorzystania współbieżności zapewnianej przez wielordzeniowe i wielosystemowe systemy. Ponadto istnieją dwa inne obszary ewolucji oprogramowania, które szybko rozwijają się w celu wsparcia paradygmatu chmury: jeden to poprawa dojrzałości i możliwości oprogramowania do zarządzania maszynami wirtualnymi, a drugi to migracja z monolitycznego podejścia w konstruowaniu rozwiązań programowych do podejścia usługowe, w którym złożone procesy składają się z luźno powiązanych komponentów. Te ostatnie kroki w ewolucji modeli sprzętu i oprogramowania doprowadziły do tego, że Grid i Cloud Computing stały się paradygmatami obniżającymi koszty rozwiązań programowych. Wykorzystywanie współdzielonych zasobów obliczeniowych z organizacji federacyjnych do wykonywania aplikacji jest kluczową koncepcją Grid Computing, zgodnie z propozycją Fostera, Kesselmana i Tuecke: „Koncepcja Grid to skoordynowane współdzielenie zasobów i rozwiązywanie problemów w dynamicznych, wieloinstytucjonalnych organizacjach wirtualnych... udostępnianie jest z konieczności ściśle kontrolowane, a dostawcy zasobów i konsumenci definiują jasno i dokładnie, co jest udostępniane, komu wolno udostępniać i w jakich warunkach następuje udostępnianie”. Ewolując z technologii Grid Computing, Utility Computing to „model biznesowy, w którym zasoby obliczeniowe są pakowane jako usługi pomiarowe” w celu spełnienia wymagań dotyczących zasobów na żądanie. Mierzone zużycie zasobów jest podobne do zużycia energii elektrycznej i wody w dostawie, a model płatności to płatność zgodnie z rzeczywistym użyciem. Projekty Utility Computing również wprowadziły i zademonstrowały idee dynamicznego udostępniania zasobów obliczeniowych. Armbrust i in. zdefiniował Cloud Computing jako dostarczanie oprogramowania aplikacyjnego dostarczanego jako usługi przez Internet oraz infrastruktury oprogramowania i sprzętu w centrach danych, które świadczą

te usługi przy użyciu modeli biznesowych podobnych do Utility Computing. Pomiar usług w celu obsługi modeli biznesowych typu pay-as-you-go jest odpowiedni dla oprogramowania aplikacyjnego (tj. oprogramowanie jako usługi, SaaS), platformy (tj. platforma jako usługi, PaaS) i infrastruktury (tj. infrastruktura jako usługi, IaaS). Co więcej, Cloud Computing wykorzystuje nowe technologie, takie jak Web 2.0 dla usług aplikacyjnych oraz wsparcie wirtualizacji i dynamicznego udostępniania usług platformowych. Grid Computing, Utility Computing i Cloud Computing różnią się między sobą pod względem architektury, rodzajów koordynowanych instytucji, rodzajów udostępnianych zasobów, modeli kosztów/biznesu oraz technologii stosowanych do osiągnięcia ich celów. Jednak wszystkie te środowiska komputerowe mają wspólne cele polegające na wykorzystaniu współdzielonych zasobów, aby optymalnie sprostać różnorodnym wymaganiom w sposób oszczędny i terminowy. Ponieważ Grid Computing rozpoczął swoją technologiczną podróż około dekadę wcześniej niż Cloud Computing, czy są lekcje do nauczenia się i technologie, które można wykorzystać z Grid do Cloud? Chcielibyśmy zbadać doświadczenia zdobyte w Grid oraz rolę technologii Grid w przetwarzaniu w chmurze. Pozostała część jest zorganizowana w następujący sposób:

- Wstępna dyskusja na temat podstaw przetwarzania w sieci i w chmurze oraz ich odpowiednich modeli interakcji między klientem, konsumentem zasobów i dostawcą
- Porównanie kluczowych procesów w Grid i Cloud computing
- Podstawowe technologie gridowe i ich zastosowanie w chmurze obliczeniowej
- Uwagi końcowe dotyczące przyszłych kierunków Grid i Cloud computing

### **Podstawy obliczeń siatkowych**

Grid Computing wykorzystuje rozproszone zasoby z różnych instytucji (dostawców zasobów), aby sprostać wymaganiom korzystających z nich klientów. Zasoby od różnych dostawców mogą być zróżnicowane i niejednorodne pod względem funkcji (przetwarzania, przechowywania, oprogramowania itp.), architektury sprzętowych (Intel x86, IBM PowerPC itp.) oraz zasad użytkowania ustalonych przez instytucje będące właścicielami. Opracowane pod parasolem Grid Computing, usługi informacyjne, usługi nazw i usługi pośrednictwa zasobów są ważnymi technologiami odpowiedzialnymi za agregację informacji o zasobach i dostępności, wybór zasobów w celu spełnienia określonych wymagań klientów i kryteriów jakości usług przy jednoczesnym przestrzeganiu zasady użytkowania zasobów. Klienci lub użytkownicy przesyłają żądania wykonania aplikacji wraz z wymaganiami dotyczącymi zasobów ze swoich domen domowych. Broker zasobów wybiera domenę z odpowiednimi zasobami do pozyskania i wykonania aplikacji lub przekierowuje aplikację do domeny w celu wykonania z wynikami i stanem z powrotem do domeny macierzystej.

### **Podstawy przetwarzania w chmurze**

IDC1 zdefiniował dwa specyficzne aspekty Chmury: Usługi w Chmurze i Przetwarzanie w Chmurze. Cloud Services to „produkty, usługi i rozwiązania konsumenckie i biznesowe, które są dostarczane i wykorzystywane w czasie rzeczywistym przez Internet”, podczas gdy Cloud Computing to „powstający model rozwoju, wdrażania i dostarczania IT, umożliwiający dostarczanie w czasie rzeczywistym produktów, usług i rozwiązania przez Internet (tj. udostępnianie usług w chmurze)”. W tym rozdziale skupimy się na infrastrukturze obliczeniowej i aspektach platformy chmury. Rozwiązanie Elastic Compute Cloud2 firmy Amazon spopularyzowało model przetwarzania w chmurze, zapewniając klientom lub użytkownikom udostępnianie na żądanie zwirtualizowanych zasobów obliczeniowych w postaci usług pomiarowych. Choć nie jest to ograniczone, większość klientów to indywidualni użytkownicy, którzy nabywają niezbędne zasoby do własnego użytku za pośrednictwem interfejsów

API EC2 bez umów lub umów międzyorganizacyjnych. Wraz z ewolucją modeli chmury wiele z nich opracowuje hybrydowy model chmury, w którym brokerzy zasobów korporacyjnych mogą pozyskiwać dodatkowe potrzebne zasoby od zewnętrznych dostawców chmury, aby sprostać wymaganiom przesłanych obciążeń korporacyjnych (E1) i żądań pracy klientów (E2). Co więcej, domena zasobów przedsiębiorstwa i dostawcy chmury mogą należeć do jednej korporacji, tworząc w ten sposób model chmury prywatnej.

### **Interakcyjne modele przetwarzania w sieci i w chmurze**

Jednym z najbardziej skalowalnych modeli interakcji w domenach Grid jest peer-to-peer, w którym większość organizacji uczestniczących w Grid to zarówno konsumenci, jak i dostawcy. W praktyce między rówieśnikami zwykle występują umowy o współdzieleniu zasobów. Ponadto klienci organizacji konsumenckich w sieciach Grid używają heterogenicznych zasobów od więcej niż jednego dostawcy zasobów należących do tej samej organizacji wirtualnej (VO) do wykonywania swoich aplikacji. Ważne jest, aby uczestniczący dostawcy zasobów i konsumenci posiadali wspólne modele informacji, protokoły interakcji, stany wykonania aplikacji itp. Organizacja Open Grid Forum (OGF)<sup>3</sup> ma na celu ustanowienie odpowiednich i niezbędnych standardów przetwarzania w sieciach grid. Niektóre proponowane standardy obejmują JSDL (Job Submission Description Language), Basic Execution Service (BES) i inne. Obecnie większość dostawców usług w chmurze oferuje własne zastrzeżone protokoły usług i formaty informacji. Ponieważ przetwarzanie w chmurze staje się dojrzałe i powszechnie stosowane, klienci i organizacje konsumenckie prawdopodobnie będą wchodzić w interakcje z więcej niż jednym dostawcą z różnych powodów, w tym w poszukiwaniu najbardziej opłacalnych rozwiązań lub nabywaniu różnych usług od różnych dostawców (np. dostawców obliczeniowych lub dostawców danych). Konsumenci chmury prawdopodobnie będą wymagać wspólnych protokołów i standardowych formatów informacji w celu ułatwienia sfederowanego użytkownika i interoperacyjności. Format Open Virtualization (OVF) grupy zadaniowej ds. zarządzania rozproszonego (DMTF)<sup>4</sup> stanowi przykładową propozycję w tym kierunku. Wzorując się na podobnych formacjach społeczności Grid, firma OGF oficjalnie uruchomiła grupę roboczą o nazwie Open Cloud Computing Interface Working Group (OCCI-WG)<sup>5</sup> w celu opracowania niezbędnych wspólnych interfejsów API do zarządzania cyklem życia usług infrastruktury chmury.

### **Przetwarzanie rozproszone w sieci i chmurze**

Grid obejmuje dwa obszary działania systemu rozproszonego. Jeden z nich ma na celu administrowanie i zarządzanie interoperacyjnym zbiorem rozproszonych klastrów zasobów obliczeniowych, na których wykonywane są zadania klienta, zazwyczaj aplikacje naukowe/HPC. Procedury i protokoły wymagane do obsługi klientów korzystających ze złożonych usług opartych na komponentach rozproszonych, które obsługują przesyłanie zadań, zabezpieczenia, udostępnianie maszyn i przemieszczanie danych. Chmura ma podobne wymagania operacyjne do obsługi złożonych usług, aby zapewnić klientom usługi na różnych poziomach wsparcia, takich jak aplikacja, platforma i infrastruktura. Grid reprezentuje również jako spójną całość zbiór zasobów obliczeniowych, które mogą znajdować się w różnych domenach administracyjnych, takich jak uniwersytety, ale współdziałać w sposób przejrzysty, tworząc wirtualne organizacje. Chociaż interoperacyjność nie jest priorytetem krótkoterminowym, istnieje precedens, by komercyjne chmury podążały w tym kierunku, podobnie jak przedsiębiorstwa, takie jak energia czy komunikacja, zawierają kontrakty ze swoimi konkurentami, aby zapewnić przepięnienie. Drugim aspektem przetwarzania rozproszonego w sieci Grid jest to, że same zadania są rozproszone, zwykle działające na ściśle powiązanych węzłach w klastrze i wykorzystujące usługi oprogramowania pośredniego, takie jak MPICH. Zadania uruchomione w siatce nie są zazwyczaj interaktywne, a niektóre mogą być częścią bardziej złożonych usług, takich jak przepływy pracy e-science. Obciążenia w chmurze zwykle składają się z luźniej powiązanych zadań rozproszonych, takich jak map/reduce, oraz zadań HPC

napisanych w celu zminimalizowania komunikacji między węzłami i wykorzystania współbieżności zapewnianej przez duże węzły wielordzeniowe. Instancje usług, które tworzą elementy większego przepływu pracy procesów biznesowych, prawdopodobnie zostaną wdrożone w chmurze. Te aspekty obciążenia zadań działających w chmurze lub sieci mają wpływ na strukturę usług, które administrują i zarządzają jakością ich wykonywania.

### **Modele warstwowe i wzorce użytkowania w siatce i chmurze**

Istnieje wiele podobieństw w systemach Grid i Cloud computing. Porównujemy podejścia, rozróżniając trzy warstwy abstrakcji w Grid: Infrastructure, Platform i Application. Następnie mapujemy te trzy warstwy na usługi Cloud IaaS, PaaS i SaaS.

#### **Infrastruktura**

Jest to warstwa, w której chmury mają większość cech wspólnych z pierwotnym przeznaczeniem oprogramowania pośredniczącego Grid. Niektóre przykłady to Eucalyptus, OpenNebula7 lub Amazon EC2. W tych systemach użytkownicy mogą udostępniać środowiska wykonawcze w postaci maszyn wirtualnych za pośrednictwem interfejsów, takich jak API lub narzędzia wiersza poleceń. Czynność definiowania środowiska wykonawczego i wysyłania żądania do końcowego zasobu ma wiele podobieństw z planowaniem zadania w Grid. Poniżej omówiono główne kroki, wspólne dla wszystkich wymienionych środowisk Cloud. Używamy Globus jako referencyjnej technologii Grid.

- Użytkownik musi mieć uprawnienia do korzystania z systemu. W systemach Grid jest to zarządzane poprzez Community Authorization System (CAS) lub poprzez kontakt z Urzędem Certyfikacji, któremu ufa instytucja docelowa, która wystawia ważny certyfikat. Chmury zazwyczaj oferują formularze internetowe umożliwiające rejestrację nowych użytkowników oraz posiadają dodatkowe aplikacje internetowe do utrzymywania baz danych klientów i generowania danych uwierzytelniających, tak jak w przypadku Eucalyptus czy Amazon.
- Gdy użytkownik ma możliwość uwierzytelnienia, musi skontaktować się z bramą, która może go zweryfikować i przetworzyć jego żądanie. Do realizacji żądań użytkowników wykorzystywane są różne mechanizmy, ale najczęstsze z nich są usługi sieciowe. Użytkownicy albo piszą niestandardowy program, który wykorzystuje theWS oferowany przez dostawców, albo korzystają z dostępnych narzędzi. Przykłady obejmują narzędzia Amazon API dla Amazon EC2, euca2ools dla Eucalyptus lub interfejs wiersza poleceń OpenNebula. Podobnie Globus oferuje zestaw skryptów opartych na konsoli, które ułatwiają komunikację z Gridem.
- W ramach żądania użycia zasobów użytkownicy muszą określić akcję lub zadanie do wykonania na zasobach docelowych. W tym celu dostępnych jest kilka formatów. Globus obsługuje język RSL (Resource Specification Language) i język JSDL (Job Submission Description Language), które mogą definiować, jaki proces ma być uruchomiony na maszynie docelowej, a także dodatkowe ograniczenia, które mogą być użyte przez komponent kojarzący w celu ograniczenia klasy zasoby, które należy wziąć pod uwagę, w oparciu o architekturę maszyny, szybkość procesora, ilość pamięci itp. Alternatywnie chmury wymagają różnych atrybutów, takich jak rozmiar środowiska wykonawczego lub obraz maszyny wirtualnej, który ma zostać użyty.
- Po otrzymaniu żądania wykonania zadania lub utworzenia środowiska następuje faza dopasowywania i planowania. Komponent GRAM firmy Globus jest pod tym względem szczególnie elastyczny, a wiele adapterów umożliwia różne traktowanie zadań: na przykład najprostszy menedżer zadań po prostu wykonuje wywołanie rozwidlenia, aby zainicjować nowy proces na docelowej maszynie. Bardziej zaawansowane i powszechnie stosowane adaptory przenoszą odpowiedzialność za

wykonanie zadań na lokalnego menedżera zasobów, takiego jak Condor, LoadLeveler lub Sun Grid Engine. Systemy te są w stanie multipleksować zadania wysyłane do ośrodka do wielu zasobów. Systemy chmurowe mają prostsze strategie zarządzania zadaniami, ponieważ rodzaje zadań są jednorodne i nie muszą być dostosowywane do różnych zasobów, takich jak w przypadku sieci Grid. Na przykład Eucalyptus wykorzystuje technikę planowania Round Robin, aby przełączać się między maszynami. OpenNebula wdraża politykę planowania rang, aby wybrać najbardziej odpowiedni zasób dla żądania i obsługuje bardziej zaawansowane funkcje, takie jak wcześniejsze rezerwacje za pośrednictwem Haizea.

- Jedną z typowych faz związanych z przesyłaniem zadań jest przesyłanie niezbędnych danych do iz maszyny wykonawczej. Pierwsza z nich, zwykle nazywana etapem, polega na pobraniu danych wyjściowych dla procesu ze zdalnego miejsca docelowego, takiego jak serwer GridFTP. Gdy ilość danych jest duża, do przetłumaczenia nazwy pliku logicznego na lokalizację można użyć usługi mapowania, takiej jak usługa lokalizacji replik (RLS). Druga część procesu, etapowa, polega albo na przeniesieniu danych wyjściowych na maszynę użytkownika, albo na umieszczeniu ich w repozytorium, ewentualnie za pomocą RLS. W przypadku przetwarzania w chmurze najważniejszymi danymi, które należy przesłać, jest definicja środowiska wykonawczego, najczęściej w postaci obrazów maszyn wirtualnych. Użytkownicy przesyłają dane opisujące system operacyjny i pakiety potrzebne do utworzenia wystąpienia maszyny wirtualnej, a następnie odwołują się do niej w celu wykonania operacji, takich jak uruchamianie nowej maszyny. Nie ma standardowej metody przesyłania danych w systemach Cloud, ale warto zwrócić uwagę na rozwiązanie obiektowej pamięci masowej Amazon, Simple Storage Service (S3), które umożliwia użytkownikom przenoszenie jednostek o rozmiarze od 1 bajta do 5 GB.

- Wreszcie, systemy Grid i Cloud muszą oferować użytkownikom metodę monitorowania ich zadań, a także wykorzystania zasobów. Ta funkcja może być również wykorzystywana przez administratorów witryny do wdrażania rozliczania użytkownika w celu śledzenia wykorzystania zasobów i egzekwowania limitów użytkowników. W kontekście Globusa można do tego celu wykorzystać dwa moduły, pierwszym jest sam GRAM, który pozwala użytkownikowi odpytywać o status wcześniej przesłanych zleceń. Drugim sposobem pozyskiwania informacji o zasobach Gridu jest usługa Monitoring and Discovery Service (MDS), która zajmuje się agregowaniem danych zasobów i udostępnianiem ich do odpytywania. Narzędzia do monitorowania wysokiego poziomu zostały opracowane w oparciu o istniejące systemy zarządzania chmurą, takie jak Amazon CloudWatch.

## **Platforma**

Ta warstwa jest zbudowana na infrastrukturze fizycznej i oferuje użytkownikom wyższy poziom abstrakcji. Interfejs zapewniany przez rozwiązanie PaaS umożliwia programistom tworzenie dodatkowych usług bez narażania się na podstawowe zasoby fizyczne lub wirtualne. Te fakty umożliwiają implementację dodatkowych funkcji w ramach modelu, takich jak prezentowanie pozornie nieskończonych zasobów użytkownikowi lub umożliwianie elastycznego zachowania na żądanie. Przykładami rozwiązań chmurowych, które prezentują te funkcje, są Google App Engine,<sup>8</sup> Salesforce's force.com<sup>9</sup> lub Microsoft Azure. W sieci Grid istnieje kilka rozwiązań, które można porównać do wspomnianych ofert PaaS, mimo że tego dokładnego modelu nie można dokładnie odtworzyć. Rozwiązania na poziomie platformy definiujemy jako te, które zawierają następujące dwa aspekty:

### **Abstrakcja z zasobów fizycznych**

Warstwa infrastruktury zapewnia użytkownikom bezpośredni dostęp do podstawowej infrastruktury. Chociaż jest to wymagane dla niższych poziomów interakcji zasobów, na poziomie Platformy użytkownik powinien być od nich odizolowany. Pozwala to programistom na tworzenie nowego

oprogramowania, które nie jest podatne na przykład na liczbę udostępnionych maszyn lub ich konfigurację sieciową.

### **Programowanie API do obsługi nowych usług**

Warstwa platformy umożliwia programistom tworzenie nowego oprogramowania, które wykorzystuje dostępne zasoby. Wybór API ma bezpośredni wpływ na programy, które można zbudować w chmurze, dlatego każde rozwiązanie PaaS jest zwykle projektowane z myślą o typie aplikacji. Dzięki tym cechom systemy Grid umożliwiają programistom tworzenie nowego oprogramowania, które wykorzystuje współdzielone zasoby w celu porównania ich z rozwiązaniami PaaS.

- Biblioteki są dostarczane przez oprogramowanie pośredniczące Grid w celu programowego dostępu do zasobów. Przykładem jest zestaw Globus Java Commodity Grid (CoG). Zestaw CoG umożliwia programistom dostęp do funkcji Grid z wyższego poziomu. Jednak zasoby muszą być adresowane niezależnie, co sprawia, że programy są powiązane z miejscami docelowymi. Dodatkowo jest powiązany z Globusem i uzależnia aplikacje od konkretnego oprogramowania pośredniczącego.
- SAGA i DRMAA11 to standardy wyższego poziomu, których celem jest zdefiniowanie niezależnego od platformy zestawu operacji Grid. skupia się na wysyłaniu i monitorowaniu zadań. Rozwiązania te zapewniają wyższy poziom abstrakcji niż w poprzednim przykładzie, ale nadal są powiązane z koncepcją zadań Grid jako programy przesyłane do zasobów zdalnych.
- Przykładem API, które omija podstawowy model Grid, aby zaoferować programistom inny paradygmat tworzenia nowego oprogramowania, jest MPICH-G2. Składa się z biblioteki, którą można połączyć z programem, który używa interfejsu przekazywania komunikatów (MPI), aby w przejrzysty sposób umożliwić aplikacji pracę w sieci Grid. Programista może myśleć w znanych terminach, nawet jeśli aplikacje obsługują sieć Grid.
- GridSuperscalar to paradygmat programowania umożliwiający uruchamianie aplikacji w sieci Grid. Programiści identyfikują funkcje swojego kodu, które mogą być uruchamiane na zdalnych zasobach, następnie określają zależności danych dla każdej z tych funkcji, a po napisaniu kodu moduł wykonawczy określa zależności danych i umieszcza każde z zadań w Grid, przysyłając dane odpowiednio, aby każde zadanie mogło zostać wykonane.
- Innym paradygmatem programowania mającym na celu budowanie nowej funkcjonalności na szczycie Gridu jest SWIFT (Zhao i in., 2007). Zapewnia język do definiowania obliczeń i zależności danych i jest specjalnie zaprojektowany do wydajnego uruchamiania bardzo dużej liczby zadań, jednocześnie ułatwiając zadanie definiowania kolejności wykonywania lub umieszczania danych generowanych z jednego zadania do wykorzystania przez inne.

Prawdopodobnie główną różnicą w paradygmatach Cloud PaaS w porównaniu z opisanymi opcjami jest to, że modele Grid muszą używać najniższego wspólnego mianownika podczas wdrażania nowych usług. Powodem tego jest to, że stopień kompatybilności z oprogramowaniem pośredniczącym jest bezpośrednio związany z liczbą dostępnych zasobów: jeśli usługa użytkownika nie przyjmie żadnych założeń co do zdalnych zasobów, będzie mogła korzystać ze wszystkich w normalny sposób; z drugiej strony usługi wymagające zainstalowania dodatkowego oprogramowania na maszynach docelowych miałyby znacznie mniej kandydatów do wykonania. W przypadku Chmur ten wymóg nie jest tak rygorystyczny z dwóch głównych powodów: po pierwsze, rozwiązania PaaS są ściśle powiązane z dostawcami Chmury, a zatem są projektowane razem z resztą infrastruktury, po drugie, że aprowizowanie zasobów z wymaganymi bibliotekami jest znacznie łatwiejsze niż w przypadku przetwarzania sieciowego, co pozwala na tworzenie nowych węzłów w wymaganym środowisku. W

przypadku Gridów, posiadanie wymaganego oprogramowania zainstalowanego w zasobach wykonawczych zwykle wiąże się z koniecznością wykonania tego przez operatora, co powoduje, że proces jest bardziej kosztowny.

## **Aplikacje**

Nie ma wyraźnych rozróżnień między aplikacjami opracowanymi w sieciach Grid a tymi, które wykorzystują chmury do wykonywania i przechowywania. Wybór platformy nie powinien wpływać na wynik końcowy, ponieważ obliczenia delegowane do systemów bazowych mogą przybierać różne kształty, aby dostosować się do dostępnych interfejsów API i zasobów. Z drugiej strony nie można zaprzeczyć, że zdecydowana większość aplikacji gridowych należy do sfery oprogramowania naukowego, podczas gdy oprogramowanie działające w chmurze skłania się ku obciążeniom komercyjnym. Tutaj staramy się zidentyfikować kilka możliwych przyczyn różnych poziomów przyjęcia tych technologii do rozwoju aplikacji:

- Brak możliwości biznesowych w Gridach. Zwykle oprogramowanie pośredniczące Grid jest instalowane tylko na sprzęcie przeznaczonym do użytku naukowego. Zjawisko to nie stworzyło pomyślnie możliwości biznesowych, które mogłyby być wykorzystane przez przemysł. I odwrotnie, chmury są zwykle wspierane przez branżę, która ma lepsze sposoby zarabiania na swoich inwestycjach.
- Złożoność narzędzi siatki. Być może ze względu na cel, jakim jest dostarczenie ustandaryzowanego, uniwersalnego rozwiązania, oprogramowanie pośredniczące Grid jest postrzegane przez wielu jako złożone i trudne do zainstalowania i zarządzania. Z drugiej strony infrastruktury chmurowe są zwykle opracowywane przez dostawców w celu dopasowania do potrzeb ich organizacji i konkretnego celu, dzięki czemu są łatwiejsze w użyciu i zorientowane na rozwiązania.
- Powiązanie z oprogramowaniem docelowym. Większość oprogramowania Grid jest opracowywana z myślą o aplikacjach naukowych, co nie jest prawdą w przypadku większości systemów chmurowych. Programy naukowe muszą uzyskiwać największą wydajność z zasobów wykonawczych, a wielu z nich nie można wydajnie uruchamiać w chmurze, na przykład z powodu narzutu wirtualizacji. Chmury są bardziej ukierunkowane na aplikacje internetowe. Te różne powinowactwa do różnych paradygmatów sprawiają, że oba rozwiązania są szczególnie skuteczne w ich docelowych zastosowaniach.

## **Techniki**

Tutaj omawiamy wpływ technik stosowanych w obliczeniach gridowych, które można zastosować w chmurach. Od czasu wprowadzenia koncepcji Grid trzeba było rozwiązać wiele problemów, aby umożliwić jej szerokie zastosowanie. Niektóre przykłady to interfejs użytkownika, transfer danych, monitorowanie zasobów lub bezpieczeństwo. Te podstawowe techniki uruchamiania sieci Grid zostały zaprojektowane, aby spełnić główne cele, a mianowicie umożliwić współdzielenie heterogenicznych zasobów między osobami należącymi do zdalnych domen administracyjnych. Cele te wyznaczają obszary zastosowania opisanych technik w Chmurach, dlatego najcenniejszy zestaw ulepszeń znajdziemy w zakresie interoperacyjności Chmury. Chmury mogą korzystać nie tylko z najbardziej podstawowych technik w Grid Computing: dodatkowe techniki, które powstały na bazie tych elementów budulcowych, aby wprowadzić nową funkcjonalność do Gridów, są również dobrymi kandydatami do zastosowania w chmurach. Wśród nich możemy znaleźć Autonomic Computing, Grid Scheduling, interoperacyjność czy symulację. Techniki omówione w tej sekcji są zatem rozłożone na różnych poziomach architektury Grid: niektóre z nich można znaleźć w niższych warstwach, udostępniając wspólne usługi innym komponentom, a inne są zbudowane z tych pierwszych i rozszerzają je. Zgodnie z klasyfikacją stwierdzamy, że niektóre techniki należą do warstwy

infrastruktury, to znaczy mają za główny cel zarządzanie zasobami, a inne są rozłożone na warstwy infrastruktury i platformy, na przykład techniki metaschedulingu opisane w sekcji planowania.

### **Orientacja na usługi i usługi internetowe**

Chmura jest zarówno dostawcą usług (np. IaaS, PaaS i SaaS), jak i miejscem do hostowania usług w imieniu klientów. Aby wdrożyć poprzednie aspekty operacyjne przy zachowaniu elastyczności, funkcje administracyjne Cloud powinny być zbudowane z komponentów oprogramowania. Grid musiał stawić czoła podobnym wyzwaniom, budując rozproszoną infrastrukturę, aby wspierać i rozwijać swoje funkcje administracyjne, takie jak bezpieczeństwo, przesyłanie zadań i tworzenie wirtualnych organizacji. Zasadą architektoniczną przyjętą przez Grid jest orientacja na usługi (SO) z komponentami oprogramowania połączonymi usługami sieciowymi (WS). W tej sekcji podsumowano wkład Open Grid Forum (OGF) w SO w zakresie przetwarzania rozproszonego oraz ich zastosowanie do chmury. SO jako architektura i usługi sieciowe jako mechanizm komunikacji między komponentami są tutaj badane w kontekście podobieństw między wymaganiami Grid i Cloud. Projektanci sieci zdali sobie sprawę z zalet luźno powiązanego modelu klienta i usług, które są odpowiednio wdrażane w rozproszonych środowiskach obliczeniowych. Pierwotnym podejściem Grid do SO była Open Grid Services Infrastructure (OGSI). OGSI został zbudowany na szczycie pojawiających się standardów usług sieci Web służących do wyrażania interfejsów między komponentami w sposób neutralny językowo w oparciu o schematy XML. Podczas gdy WS jest interfejsem, OGSI próbowało uczynić go zorientowanym obiektowo, dodając wymagane metody. Następnie społeczność Grid pracowała w ramach standardów WS, aby rozszerzyć specyfikację WS w oparciu o doświadczenia z wykorzystaniem SOA. Doprowadziło to do wprowadzenia architektury Open Grid Services Architecture (OGSA), zaimplementowanej w wersji 3 zestawu narzędzi Globus. OGSA zawiera kluczowe rozszerzenia standardu WS, które są teraz opisane. W Grid i Cloud najbardziej typowe składniki usług, takie jak udostępnianie obrazu systemu operacyjnego, uruchamianie maszyny wirtualnej lub wysyłanie zadania, działają długo. Kompozycja usług tych składników wymaga asynchronicznego modelu programowania. Komponent usługi konsumenckiej wywołuje dostawcę WS i jest natychmiast potwierdzany, więc dzwoniący nie utrzymuje otwartego procesu na łączu komunikacyjnym. Komponent dostawcy asynchronicznie aktualizuje konsumenta w miarę zmiany stanu. Architekci sieci uznali znaczenie obsługi modelu asynchronicznego i zintegrowali to podejście z usługami sieci Web za pośrednictwem rozszerzeń WS-Addressing i WS-Notify. WS-Addressing określa sposób odwoływania się nie tylko do punktów końcowych usługi, ale także do obiektów w punkcie końcowym usługi. Powiadomienie opiera się na WS-Addressing, który określa składnik, który ma być powiadamiany o zmianie stanu. Związane z długowiecznym działaniem usługi i komunikacją asynchroniczną to wymóg utrzymywania i udostępniania informacji o stanie. Istnieje wiele sposobów na osiągnięcie stanu, żaden prosty, zwłaszcza gdy wiele usług może aktualizować ten sam obiekt stanu. W zasadzie interfejs WS jest bezstanowy, chociaż oczywiście istnieje wiele sposobów budowania aplikacji na bazie WS, które przekazują stan przez komunikaty operacji. Wyzwaniem jest zintegrowanie specyfikacji WS ze standardem stanu, który nie zakłóca bezstanowej intencji modelu interfejsu WS. OGF osiągnął ten cel, rozwijając Web Service Resource Framework (WSRF). WSRF umożliwia metodom fabrycznym w implementacji WS tworzenie obiektów, do których odwołuje się zdalnie przy użyciu standardu WS-Addressing. Trwałe właściwości zasobów są udostępniane połączonym usługom za pośrednictwem XML. Wprowadzenie stanu potencjalnie zwiększa złożoność systemu rozproszonego i dystrybucja danych stanowych do wielu komponentów usług może potencjalnie powodować problemy ze spójnością danych, które wymagałyby rozproszonych mechanizmów blokowania. Podejście wprowadzone przez Grid przekazuje punkty końcowe WS do zasobów, dzięki czemu zsynchronizowany dostęp jest zapewniany przez implementację usługi. Jednym ze sposobów wykorzystania doświadczeń technologii Grid w chmurze jest rozważenie usługi wsparcia eksploatacji budynku z SO. Usługi składowe łączą się za pomocą



zestawu standardów WS. Logika tworzenia usług jest zbudowana przy użyciu nowoczesnych narzędzi do projektowania procesów biznesowych, które tworzą przepływ pracy. Przepływ pracy projektu jest eksportowany w postaci języka BPEL (Business Process Execution Language) i wykonywany przez silnik przepływu pracy. Ta ścieżka implementacji polegająca na użyciu BPEL z WSRF do budowania architektury SOA została zademonstrowana w kontekście e-Science. Istnieje już pewne doświadczenie w korzystaniu z WS i WSRF w domenie Cloud. Projekt Nimbus wykorzystuje model WS i WSRF jako interfejs umożliwiający klientom dostęp do obszarów roboczych w chmurze.

## **Zarządzanie danymi**

W obliczeniach siatkowych aplikacje intensywnie wykorzystujące dane, takie jak oprogramowanie naukowe w dziedzinach takich jak fizyka wysokich energii, bioinformatyka, astronomia lub nauki o Ziemi, obejmują duże ilości danych, czasami w skali petabajtów (PB) i dalej. Techniki zarządzania danymi służące do wykrywania i uzyskiwania dostępu do informacji są niezbędne dla tego rodzaju aplikacji. Przepustowość sieci, opóźnienia transferu i zasoby pamięci masowej są tak samo ważne jak zasoby obliczeniowe, aby określić opóźnienia i wydajność zadań. Na przykład aplikacja intensywnie przetwarzająca dane będzie najlepiej uruchamiana w witrynie, która ma obszerny i szybki kanał sieciowy do swojego zestawu danych, dzięki czemu można zmniejszyć obciążenie sieci, a jeśli generuje dużą ilość danych, wolelibyśmy również miejsce, które ma wystarczająco dużo miejsca do przechowywania w pobliżu. Wiele technologii jest stosowanych w przetwarzaniu sieciowym w celu rozwiązania problemów związanych z zarządzaniem danymi. Siatki danych pojawiły się w środowisku naukowym i komercyjnym, aby w szczególności zoptymalizować zarządzanie danymi. Na przykład jedną z usług świadczonych przez usługi Data Grids jest zarządzanie replikami. Aby skutecznie pobierać dane, a także unikać gorących punktów w środowisku rozproszonym, siatki danych często przechowują repliki, które są albo pełnymi, albo częściowymi kopiami oryginalnych zestawów danych. Usługi zarządzania replikami są odpowiedzialne za tworzenie i rejestrowanie replik oraz zarządzanie nimi. Zwykle tworzony i zarządzany jest katalog replik, taki jak Globus Replica Catalog, zawierający informacje o replikach, które mogą być zlokalizowane przez użytkowników. Oprócz replikacji danych buforowanie jest skuteczną metodą zmniejszania opóźnień po stronie konsumenta danych. W siatkach danych stosowane są również inne technologie, takie jak przesyłanie strumieniowe, wstępne przygotowywanie, szybkie przenoszenie danych lub optymalny dobór źródeł i ujścia danych. Te technologie zarządzania danymi są również wykorzystywane w systemach udostępniania i dystrybucji danych, takich jak sieci dostarczania treści, sieci peer-to-peer i rozproszone bazy danych. W Venugopal, Buyya i Ramamohanarao autor proponuje taksonomię siatek danych i porównuje siatki danych z innymi powiązаныmi obszarami badawczymi. W społeczności Grid zaproponowano standardy usług danych. Architektura Open Grid Services Architecture (OGSA), przyjęta przez Global Grid Forum (GGF), definiuje usługi danych OGSA, które obejmują transfer danych, dostęp do danych, zarządzanie zasobami pamięci masowej, pamięć podręczną danych, replikację danych, federację danych i usługi katalogów metadanych. Grupa Robocza ds. Dostępu i Usług Integracji Baz Danych (DAIS-WG) w GGF opracowuje również standardy usług danych z naciskiem na systemy zarządzania bazami danych, które odgrywają centralną rolę w zarządzaniu danymi, takimi jak przechowywanie danych, dostęp, organizacja, autoryzacja itp. Istnieją inne grupy w GGF, które pracują nad zarządzaniem danymi w Grid Computing, takie jak Grid File System Working Group, Grid Storage Management Working Group lub GridFTP Working Group. Wśród nich GridFTP Working Group pracuje nad poprawą wydajności FTP i GridFTP. GridFTP jest rozszerzeniem FTP i obsługuje równoległy i rozłożony transfer danych oraz częściowy transfer plików. FTP i GridFTP to najczęściej używane protokoły transportowe podczas przesyłania masowych danych dla aplikacji Grid. Globus Toolkit zapewnia wiele rozwiązań do zarządzania danymi, w tym GridFTP, globalny dostęp do pamięci dodatkowej (GASS), niezawodny transfer plików (RFT), usługę lokalizacji replik (RLS) oraz usługę replikacji danych wyższego poziomu

(DRS) opartą na RFT i RLS. W szczególności GASS to lekki mechanizm dostępu do danych dla zdalnych systemów pamięci masowej. Umożliwia wstępne i końcowe umieszczanie plików i jest zintegrowane z Globus Resource Access and Monitoring (GRAM) w celu przygotowania plików wykonywalnych i danych wejściowych oraz, jeśli to konieczne, wystawienia danych wyjściowych i dzienników. W obecnym stanie przetwarzania w chmurze przechowywanie jest zwykle zbliżone do obliczeń, a zatem zarządzanie danymi jest prostsze niż w sieciach Grid, gdzie pula zasobów wykonawczych i pamięci masowej jest znacznie większa, a zatem do umieszczania zadań i lokalizacji danych wymagane są wydajne i skalowalne metody i transfer. Mimo to należy wziąć pod uwagę dostęp do danych, aby zapewnić lepszą wydajność aplikacji. Przykładem tego jest Hadoop, który planuje obliczenia blisko danych, aby zmniejszyć opóźnienia przesyłania. Podobnie jak przetwarzanie sieciowe, chmury muszą zapewniać skalowalne i wydajne techniki do przesyłania danych. Na przykład możemy potrzebować przenieść obrazy maszyn wirtualnych, które są używane do tworzenia instancji środowisk wykonawczych w chmurach, z użytkowników do repozytorium i z repozytorium do maszyn hostujących. Techniki poprawiające szybkość transferu, takie jak GridFTP, skutkowałyby krótszymi czasami dla witryn, które mają dużą przepustowość, ponieważ mogą optymalizować transfer danych poprzez zrównoleglenie strumieni wysyłających. Ponadto usługi katalogowe można wykorzystać do poprawy rozproszonego udostępniania informacji między wieloma uczestnikami, tak aby lokalizacja danych użytkowników i repozytoriów danych była bardziej wydajna. Standardy opracowane na podstawie praktyki przetwarzania w sieciach mogą być wykorzystane do poprawy interoperacyjności wielu chmur. Wreszcie lepsza integracja zarządzania danymi z infrastrukturą bezpieczeństwa umożliwiłaby grupom zaufanych użytkowników. Zastosowanie tej zasady może być stosowane w systemach takich jak Amazon EC2, gdzie obrazy maszyn wirtualnych są udostępniane przez osoby nieposiadające zapewnienia o ich pochodzeniu.

## **Monitorowanie**

Chociaż niektóre narzędzia do monitorowania w chmurze zostały już opracowane, dostarczają informacji na wysokim poziomie, a w większości przypadków funkcja monitorowania jest osadzona w systemie VMmanagement zgodnie z określonymi mechanizmami i modelami. Obecnym wyzwaniem dla narzędzi do monitorowania Chmury jest dostarczanie informacji z Chmur i żądań aplikacji/usług z wystarczającym poziomem szczegółowości w czasie niemal rzeczywistym, aby podejmować skuteczne decyzje, zamiast zapewniać prostą i graficzną reprezentację stanu Chmury. Aby to zrobić, różne technologie monitorowania sieci mogą być stosowane do Chmur, szczególnie tych, które są w stanie dostarczać dane z monitoringu w formie zagregowanej ze względu na dużą skalę i dynamiczne zachowanie Chmur. Doświadczenia zdobyte podczas badań nad standaryzacją monitorowania sieci mogą pomóc w zdefiniowaniu ujednoczonych i standardowych interfejsów monitorowania i modeli danych w celu zwiększenia interoperacyjności między różnymi chmurami. Monitorowanie sieci to złożone zadanie, ponieważ natura sieci oznacza niejednorodne systemy i zasoby. Jednak monitorowanie jest niezbędne w sieci Grid, aby umożliwić rozliczanie wykorzystania zasobów i informowanie użytkowników, czy i jak działają ich zadania. Jest to również ważny aspekt w przypadku innych zadań, takich jak planowanie. Grupa Robocza OGF Performance opracowała model narzędzi do monitorowania sieci o nazwie Grid Monitoring Architecture (GMA). Proponowana przez nich architektura została zaprojektowana tak, aby uwzględniać cechy platform Grid. Informacje o wydajności mają stały, często krótki okres użyteczności. Dane dotyczące wydajności są często aktualizowane częściej niż żądane, podczas gdy zwykłe programy bazodanowe są najpierw przeznaczone do zapytań. Oznacza to, że trwałe przechowywanie nie zawsze jest konieczne, a narzędzia muszą być w stanie szybko odpowiedzieć, zanim dane staną się przestarzałe. Narzędzie do monitorowania wydajności sieci musi również obsługiwać wiele różnych rodzajów zasobów i powinno być w stanie dostosować się, gdy łączy komunikacyjne lub inne zasoby ulegną awarii. W związku z tym

systemy monitorowania powinny być dystrybuowane tak, aby spełniały te wymagania. W rzeczywistości narzędzie do monitorowania powinno znaleźć dobry kompromis między następującymi cechami: małym opóźnieniem dostarczania danych, dużą szybkością przesyłania danych, skalowalnością, zasadami bezpieczeństwa i minimalną ingerencją. GMA opiera się na trzech rodzajach komponentów: producentach, konsumentach i usłudze katalogowej. Producentem jest dowolny komponent, który może wysyłać zdarzenia do konsumenta za pomocą interfejsu producenta (akceptuje subskrypcję, zapytania i możliwość powiadamiania). W narzędziu monitorującym każdy czujnik jest umieszczony w obudowie producenta; jednak producent może być powiązany z wieloma różnymi źródłami: np. czujnikami, systemami monitoringu czy bazami danych. Konsument to dowolny komponent, który może odbierać dane o zdarzeniach od producenta. Interfejs konsumenta zawiera procedury subskrypcji/anulowania subskrypcji oraz mechanizmy zapytań. Aby wymieniać zdarzenia danych, producenci i konsumenci mają bezpośrednie połączenie, ale aby zainicjować dialog, potrzebują usługi katalogowej. Opracowano kilka narzędzi monitorujących dla systemów Grid. Balaton i inni przedstawili opis i kategoryzację istniejących narzędzi monitorowania i oceny wyników, a Serafeim i inni zaproponowali taksonomię systemów monitorowania sieci, która jest wykorzystywana do klasyfikowania szerokiego zakresu projektów i ram. Niektóre z tych podejść omówiono poniżej.

Ganglia to skalowalny rozproszony system monitorowania dla środowisk obliczeniowych o wysokiej wydajności, takich jak klastry i sieci grid. Opiera się na hierarchicznym projekcie ukierunkowanym na federację klastrów, opiera się na protokole nasłuchiwanie/ogłaszania opartym na multemisji do monitorowania stanu w klastrach i wykorzystuje śledzenie połączeń punkt-punkt między reprezentatywnymi węzłami klastra w celu federacji klastrów i agregowania ich stanu. Dane są reprezentowane w XML i skompresowane przy użyciu XDR. Frontend GangliaWeb może być używany do sprawdzania na przykład wykorzystania procesora w ciągu ostatniej godziny lub ostatniego miesiąca. Ganglia została wdrożona w wielu infrastrukturach HPC, w tym w superkomputerach i dużych systemach Grid.

Network Weather Service (NWS) to rozproszony system do tworzenia krótkoterminowych prognoz wydajności na podstawie historycznych pomiarów wydajności. NWS zapewnia zestaw czujników systemowych do okresowego monitorowania kompleksowej wydajności TCP/IP (przepustowość i opóźnienie), dostępnego procentu procesora i dostępnej pamięci niestroniowanej. Na podstawie zebranych danych NWS dynamicznie charakteryzuje i prognozuje wydajność zasobów sieciowych i obliczeniowych.

Mercury został zaprojektowany w celu spełnienia wymagań monitorowania wydajności sieci: dostarcza dane monitorowania reprezentowane jako metryki za pomocą semantyki dostępu typu „pull” i „push”, a także wspiera sterowanie za pomocą elementów sterujących. Obsługuje monitorowanie jednostek Grid, takich jak zasoby i aplikacje, w sposób ogólny, rozszerzalny i skalowalny. Jego konstrukcja jest zgodna z wcześniej opisanymi zaleceniami OGF GMA.

OCM-G to zgodny z OMIS monitor aplikacji opracowany w ramach projektu CrossGrid. Zapewnia konfigurowalne monitorowanie online za pośrednictwem centralnego menedżera, który przekazuje żądania informacji do lokalnego monitora. Jednak OCM-G ma architekturę rozproszoną.

Globus Monitoring and Discovery System (MDS) to kolejne szeroko stosowane narzędzie monitorujące, które dostarcza informacji o dostępnych zasobach w sieci Grid i ich statusie. Opiera się na schemacie GLUE14, który służy do zapewnienia jednolitego opisu zasobów i ułatwienia współpracy między infrastrukturami Grid. Opracowano inne podejścia do systemów wielkoskalowych, takie jak MonALISA, która jest rozszerzalną ramą monitorowania hostów i sieci w systemach rozproszonych na dużą skalę, oraz Palantir, który został zaprojektowany w celu ujednoczenia dostępu do różnych systemów

monitorowania i informacji w celu współdzielenia zasobów na dużą skalę w różnych domenach administracyjnych, zapewniając w ten sposób ogólne sposoby dostępu do wszystkich tych informacji. Ponadto różne struktury portali Grid zawierają funkcje monitorowania, takie jak pojedynczy punkt dostępu HPC-Europa i portal P-GRADE. Kilka centrów danych, które dostarczają zasoby do systemów Cloud, przyjęło Ganglia jako narzędzie do monitorowania. Jednak środowiska zwirtualizowane mają bardziej specyficzne potrzeby, które zmotywowały dostawców technologii przetwarzania w chmurze do opracowania własnego systemu monitorowania. Niektóre z nich zostały podsumowane poniżej:

Amazon CloudWatch to usługa internetowa, która zapewnia monitorowanie zasobów Amazon Web Services Cloud, takich jak Amazon EC2. Zbiera surowe dane z AmazonWeb Services, a następnie przetwarza informacje w czytelne wskaźniki, które są rejestrowane przez okres dwóch tygodni. Zapewnia użytkownikom wgląd w wykorzystanie zasobów, wydajność operacyjną i ogólne wzorce popytu – w tym metryki, takie jak wykorzystanie procesora, odczyty i zapisy na dysku oraz ruch sieciowy.

Monitor diagnostyczny systemu Windows Azure zbiera dane w magazynie lokalnym dla każdego włączonego typu diagnostyki i może przysyłać zebrane dane na konto usługi Azure Storage w celu trwałego przechowywania. Można zaplanować przesyłanie zebranych danych do pamięci w regularnych odstępach czasu lub można zażądać transferu na żądanie, gdy tylko te informacje są wymagane.

Menedżer informacji OpenNebula (IM) jest odpowiedzialny za monitorowanie różnych węzłów w chmurze. Zawiera różne czujniki, z których każdy odpowiada za różne aspekty monitorowanych zasobów obliczeniowych (procesor, pamięć, nazwa hosta). Istnieją również czujniki przygotowane do zbierania informacji z różnych hipernadzorców.

Funkcjonalność monitoringu Aneki realizowana jest przez coreware middleware, które zapewnia szeroki zestaw usług obejmujący również negocjowanie jakości usługi, kontrolę wstępu, zarządzanie realizacją, księgowość i rozliczenia. Aby pomóc administratorom w dostrojeniu ogólnej wydajności chmury, Management Studio zapewnia zagregowane dynamiczne statystyki.

Rozwiązanie Nimsoft Monitoring Solution (NMS), zbudowane w oparciu o architekturę Nimsoft Unified Monitoring Architecture, zapewnia funkcjonalność monitorowania dowolnej kombinacji zwirtualizowanego centrum danych, hostowanej lub zarządzanej infrastruktury, w chmurze na platformie IaaS lub PaaS lub dostarczanej jako usługi SaaS. W szczególności zapewnia ujednoczone monitorowanie centrów danych, chmur prywatnych i publicznych, takich jak Amazon WS, w tym monitorowanie, wizualizację i raportowanie poziomu usług i czasu reakcji.

Hyperic CloudStatus zapewnia oprogramowanie do monitorowania i zarządzania open source dla wszystkich typów aplikacji internetowych, zarówno hostowanych w chmurze, jak i lokalnie, w tym Amazon Web Services i Google App Engine. CloudStatus zapewnia użytkownikom raporty w czasie rzeczywistym i tygodniowe trendy dotyczące metryk infrastruktury.

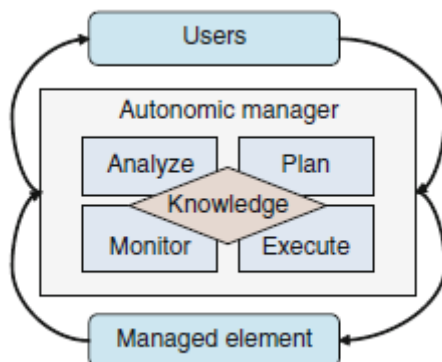
### **Obliczenia autonomiczne**

Zainspirowane autonomicznym układem nerwowym, autonomiczne przetwarzanie ma na celu projektowanie i budowanie systemów samozarządzających i stało się obiecującym podejściem do rozwiązywania problemów związanych ze złożonością oprogramowania. System autonomiczny jest w stanie podejmować decyzje dotyczące reagowania na zmiany warunków działania w czasie wykonywania, korzystając z polityk wysokiego poziomu, które są zazwyczaj dostarczane przez eksperta. Taki system stale monitoruje i optymalizuje swoje działanie oraz automatycznie dostosowuje

się do zmieniających się warunków, aby dalej osiągać swoje cele. Istnieje kilka ważnych i cennych kamieni milowych, aby osiągnąć pełną autonomię

- informatyka: po pierwsze, zautomatyzowane funkcje będą jedynie gromadzić i agregować informacje w celu wsparcia decyzji podejmowanych przez użytkowników. Później będą służyć jako doradcy, proponując ludziom możliwe kierunki działania. Samozarządzanie jest istotą autonomicznego informatyki i zostało zdefiniowane w kategoriach następujących czterech aspektów samozarządzania.
- Samokonfiguracja: systemy autonomiczne konfiguruje się automatycznie zgodnie z zasadami wysokiego poziomu reprezentującymi cele na poziomie biznesowym, które na przykład określają, co jest pożądane, a nie jak ma zostać osiągnięte. Po wprowadzeniu komponentu bezproblemowo się włączy, a reszta systemu dostosuje się do jego obecności.
- Samoptymalizacja: systemy autonomiczne będą nieustannie poszukiwać sposobów na poprawę ich działania, identyfikując i wykorzystując możliwości, aby zwiększyć wydajność i/lub koszty. Systemy autonomiczne będą monitorować, eksperymentować i dostrajać własne parametry oraz nauczą się dokonywać właściwych wyborów dotyczących zachowania funkcji lub ich outsourcingu.
- Samonaprawianie: autonomiczne systemy obliczeniowe będą wykrywać, diagnozować i naprawiać zlokalizowane problemy wynikające z błędów lub awarii oprogramowania i sprzętu.
- Samoobrona: Systemy autonomiczne będą samoobronne w dwóch znaczeniach. Będą chronić system jako całość przed skorelowanymi problemami na dużą skalę wynikającymi ze złośliwych ataków lub kaskadowych awarii, których nie można naprawić za pomocą środków samonaprawczych. Będą również przewidywać problemy na podstawie wczesnych raportów z czujników i podejmować kroki w celu ich uniknięcia lub złagodzenia.

Rysunek przedstawia jedną podstawową strukturę elementu autonomicznego zaproponowaną przez IBM.



Składa się z autonomicznego menedżera, który monitoruje, analizuje, planuje i wykonuje w oparciu o zebraną wiedzę oraz środowisk zewnętrznych obejmujących użytkowników i zarządzane elementy. Elementem zarządzanym mogą być zasoby sprzętowe, takie jak procesor, pamięć i pamięć masowa, zasoby oprogramowania, takie jak baza danych, usługa katalogowa lub system lub aplikacja. Zarządzający autonomiczny monitoruje zarządzane elementy i ich otoczenie zewnętrzne, w tym zmieniające się wymagania użytkowników, i analizuje je, oblicza nowy plan uwzględniający zmieniające się warunki i realizuje ten plan. Koncepty autonomiczne zostały skutecznie zastosowane w rozproszonych środowiskach obliczeniowych, takich jak sieci Grid i systemy komunikacji/sieci, w celu

monitorowania zasobów, zmiany obciążeń lub modeli, a następnie dostosowania udostępniania zasobów w celu spełnienia wymagań i ograniczeń. Takie możliwości zostały włączone do systemów Cloud. Używamy CometCloud jako przypadku użycia, który zapewnia autonomiczne możliwości na wszystkich poziomach. CometCloud to autonomiczny silnik obliczeniowy dla środowisk Cloud i Grid. Opiera się na zdecentralizowanym podłożu koordynacyjnym i obsługuje aplikacje autonomiczne w wysoce heterogenicznych i dynamicznych infrastrukturach Cloud/Grid, a także integrację publicznych/prywatnych chmur/sieci. Na przykład obsługuje autonomiczne cloudbursts, których celem jest bezproblemowe (i bezpieczne) połączenie chmur prywatnych przedsiębiorstw i centrów danych z chmurami lub sieciami użyteczności publicznej na żądanie, aby zapewnić abstrakcję skalowalnej mocy obliczeniowej, która jest sterowana przez zdefiniowane przez użytkownika polityk wysokiego szczebla. Umożliwia dynamiczne wdrażanie komponentów aplikacji, które zwykle działają na wewnętrznych zasobach obliczeniowych organizacji, w publicznej chmurze lub sieciach (tj. Cloudburst), aby sprostać dynamicznym obciążeniom, skokom zapotrzebowania, problemom ekonomicznym/budżetowym i innym ekstremalnym wymaganiom. Co więcej, biorąc pod uwagę rosnącą skalę aplikacji i infrastruktury, a także koszty ich chłodzenia, eksploatacji i zarządzania, typowe strategie alokacji nadmiarowej nie są już możliwe. Autonomiczne cloudbursts może wykorzystywać chmury narzędziowe, aby zapewnić możliwości skalowania w poziomie i w pionie na żądanie w oparciu o szereg metryk. Innymi przykładami technologii Clouds, które przejmują autonomiczne techniki obliczeniowe z Grid Computing, są Aneka i CloudWatch.

### **Planowanie, metaplanowanie i udostępnianie zasobów**

W ciągu ostatnich kilku dekad wiele wysiłku poświęcono badaniom nad planowaniem pracy, zwłaszcza w ośrodkach z urządzeniami High Performance Computing (HPC). Ogólny problem planowania składa się z zestawu zadań i wymagań, zestawu zasobów i stanu systemu, decydując, które zadania rozpocząć i w których zasobach. W literaturze istnieje wiele polityk planowania zadań, takich jak podejście FCFS i jego warianty. Inne polityki wykorzystują szacunkowe informacje o aplikacji (na przykład czas wykonania), które nie zakładają żadnych założeń, takich jak najmniejsze zadanie najpierw (SJF), największe zadanie najpierw (LJF), Najpierw najmniejsze skumulowane zapotrzebowanie (SCDF) lub zasypywanie, które jest jednym z najczęściej stosowanych w systemach HPC. W obliczeniach siatkowych techniki planowania ewoluowały w celu uwzględnienia innych czynników, takich jak niejednorodność zasobów lub rozmieszczenie geograficzne. Komponent oprogramowania odpowiedzialny za planowanie zadań w Grids jest zwykle nazywany meta-schedulerem lub brokerem zasobów Grid. Główne akcje wykonywane przez brokera zasobów Grid to: wykrywanie i monitorowanie zasobów, wybór zasobów, wykonywanie zadań, obsługa i monitorowanie. Może jednak odpowiadać również za inne dodatkowe zadania, takie jak mechanizmy bezpieczeństwa, księgowość, jakość obsługi (QoS) zapewnianie, rezerwacje z wyprzedzeniem, negocjacje z innymi jednostkami planowania, egzekwowanie zasad, migracja itp. Niektóre z ich najczęstszych cech omówiono w następujący sposób:

- Mogą obejmować różne warstwy planowania poprzez kilka składników oprogramowania między brokerem zasobów Grid a zasobami, na których będzie działać aplikacja. W związku z tym informacje i kontrola dostępne na poziomie brokera zasobów są znacznie mniejsze niż te dostępne na poziomie planowania klastra.
- Broker zasobów Grid zwykle nie ma prawa własności ani kontroli nad zasobami. Co więcej, systemy planowania klastrów mogą mieć własne polityki lokalne, które mogą kolidować ze strategią planowania sieci.

- Występują sprzeczne cele dotyczące wydajności między użytkownikami a właścicielami zasobów. Podczas gdy użytkownicy koncentrują się na optymalizacji wydajności pojedynczej aplikacji dla określonego celu kosztowego, właściciele zasobów dążą do uzyskania najlepszej przepustowości systemu lub zminimalizowania czasu odpowiedzi.

Podczas gdy w Grid Computing najważniejszymi zadaniami planowania są optymalizacja czasu reakcji aplikacji i wykorzystania zasobów, w Cloud computingu kluczowe stają się inne czynniki, takie jak względy ekonomiczne i wydajne dostarczanie zasobów pod względem gwarancji QoS, wykorzystania i energii. Ponieważ zwirtualizowane centra danych i chmury zapewniają abstrakcję prawie nieograniczonych zasobów obliczeniowych dzięki elastycznemu wykorzystaniu skonsolidowanych pul zasobów, zadanie planowania przenosi się na planowanie zasobów (tj. udostępnianie żądań aplikacji z zasobami). Omawiany problem z obsługą administracyjną polega na tym, jak dynamicznie alokować zasoby między maszynami wirtualnymi w celu zoptymalizowania funkcji globalnego narzędzia. Niektóre przykłady to minimalizacja nadmiernej alokacji zasobów (marnotrawstwo zasobów) i maksymalizacja QoS (aby zapobiec popadaniu w niedostateczną alokację, która może prowadzić do utraty przychodów dostawców). Zaproponowano różne techniki udostępniania centrów danych, takie jak te oparte na planowaniu gangów, oparte o rezerwacje z wyprzedzeniem, te oparte na efektywności energetycznej lub oparte na wielopoziomowym podejściu do planowania zasobów. Chociaż planowanie w chmurze obliczeniowej nadal stanowi wyzwanie, można wziąć pod uwagę kilka technik opracowanych dla środowisk Grid. Istniejące techniki planowania zadań można również zastosować w środowiskach zwirtualizowanych, zwłaszcza gdy stawki żądań aplikacji są zgodne z oczekiwaniami w przyszłych chmurach. W rzeczywistości niektóre podejścia zaczęły zajmować się tym problemem. Rezerwacja z wyprzedzeniem opracowana dla planowania sieci jest używana w menedżerze dzierżawy Haizea dla OpenNebula. Różne zasady zarządzania SLA dla Grid computing zostały rozszerzone dla chmur, takie jak te proponowane przez Buyya, Yeo, Venugopal, Broberg i Brasdic. Zorientowana na rynek polityka alokacji zasobów opracowana dla Gridów została zrealizowana dla Chmury w Anece. Sodan proponuje adaptacyjne planowanie, które może dostosowywać rozmiary zadań równoległych w celu uwzględnienia różnych sytuacji obciążenia i różnej dostępności zasobów poprzez zmianę kształtu zadań i zmianę rozmiaru maszyn wirtualnych. Co więcej, strategie planowania w chmurze obliczeniowej mogą wykorzystywać wielowarstwowe architektury Grid i strategie, takie jak międzywarstwowa polityka optymalizacji QoS zaproponowana przez Chunlin i Layuan.

### **Interoperacyjność w sieciach i chmurach**

Jednym z celów Grid Computing jest zapewnienie jednolitego i spójnego dostępu do zasobów rozproszonych w różnych centrach danych i instytucjach. Wynika to z faktu, że większość Gridów powstaje w oparciu o inicjatywy regionalne, a nie lokalne, więc interoperacyjność jest kluczowym celem. Niektóre przykłady to TeraGrid w USA, GridX1 w Kanadzie, Naregi w Japonii i EGEE w Europie. Współpraca dotyczy różnych punktów architektury, takich jak portal dostępu, funkcja pośrednictwa zasobów i standaryzacja infrastruktury. Niektóre środowiska produkcyjne Grid, takie jak HPC-Europa, DEISA i PRACE, podchodzą do interoperacyjności za pomocą jednolitego interfejsu dostępu dla użytkowników aplikacji. Warstwy oprogramowania pod interfejsem użytkownika abstrahują następnie od złożoności podstawowej heterogenicznej infrastruktury superkomputerowej. Jednym z narzędzi, które stosuje takie podejście do współdziałania sieci, jest meta-brokering. Meta-brokering wspiera interoperacyjność sieci Grid z punktu widzenia zarządzania zasobami i planowania. Wiele projektów bada to podejście z różnymi akcentami. Poniżej omówiono przykłady pogrupowane luźno według głównych ośrodków technicznych.

- Interoperacyjność infrastruktury

GridWay , który opiera się głównie na Globus, obsługuje wiele Gridów korzystających z bram Grid w celu uzyskania dostępu do zasobów należących do różnych domen. GridWay przekazuje żądania użytkowników lokalnych do innej domeny, gdy bieżąca jest przeciążona. W ramach projektu Latin American Grid Meta-brokering zaproponowano i wdrożono wspólny zestaw protokołów, aby umożliwić współdziałanie heterogenicznych meta-schematów zorganizowanych w strukturę peer-to-peer. Wybór domeny zasobów jest oparty na zagregowanym modelu informacji o zasobach, a zadania z domeny domowej mogą być kierowane do domen równorzędnych w celu wykonania.

- Optymalizacja zasobów we współdziałających sieciach

Koala Grid Scheduler koncentruje się na współlokacji danych i procesorów. Aby połączyć różne domeny Grid jako różne instancje Koala. Ich zasadą jest używanie zasobów z domeny zdalnej tylko wtedy, gdy domena lokalna jest nasycona. Używają delegowanego kojarzenia, aby uzyskać dopasowane zasoby z jednej z równorzędnych instancji Koala bez kierowania zadań do domen równorzędnych.

InterGrid promuje łączenie różnych systemów Grid poprzez umowy peeringowe oparte na podejściach ekonomicznych, aby umożliwić współdzielenie zasobów między sieciami. Jest to podejście oparte na ekonomii, w którym wsparcie aplikacji biznesowych jest podstawowym celem, a to ma również zapewnić zrównoważony rozwój.

VIOLA MetaScheduling Service implementuje interoperacyjność Grid poprzez mechanizmy SLA (WS-Agreement) i zapewnia współlokację wielu zasobów na podstawie rezerwacji.

Inne projekty badają interoperacyjność systemów Grid poprzez wykorzystanie standardowych mechanizmów, protokołów i interfejsów. Na przykład projekt Grid Interoperability Project (GRIP), projekt Open Middleware Infrastructure Institute for Europe (OMII-Europe) lub prace wykonane w ramach portalu P-GRADE. GRIP był jedną z pierwszych propozycji umożliwiających interoperacyjność między UNICORE i Globus Toolkit. Celem projektu OMII-Europe było wpłynięcie na przyjęcie i rozwój otwartych standardów, które ułatwiają interoperacyjność między gLite i UNICORE, takich jak OGSA BES lub JSDL. Portal P-GRADE próbuje połączyć różne infrastruktury Grid, zapewniając dostęp do zgodnego ze standardami, interoperacyjnego oprogramowania pośredniczącego. Grid Interoperation Now Community Group (GIN-CG) oraz Grupa Robocza ds. Infrastruktury Produkcyjnej Grid (PGI-WG) OGF również zajmują się problemem interoperacyjności sieci. W pierwszym przypadku kierowanie i weryfikację strategii interoperacyjnych, aw drugim zorientowane na produkcyjne infrastruktury gridowe. Chociaż osiągnięto znaczne postępy w zakresie interoperacyjności w Grid Computing, interoperacyjność między dostawcami chmury nie została jeszcze zbadana. Chociaż entuzjazm w tworzeniu interoperacyjności chmury jest ograniczony wśród dostawców chmury nastawionych na zysk, jest wielu zwolenników w społecznościach akademickich i naukowych. Projekt RESERVOIR dotyczy interoperacyjności chmury za pomocą modułowej, rozszerzalnej architektury chmury opartej na federacji chmur. W modelu RESERVOIR każdy dostawca infrastruktury jest autonomiczną firmą z własnymi celami biznesowymi. Dostawca łączy się z innymi dostawcami na podstawie zasad dostosowanych do celów biznesowych witryny. W kontekście RESERVOIR interfejsy i protokoły Grid mogą umożliwiać wymaganą interoperacyjność między chmurami lub dostawcami infrastruktury. Podobną inicjatywą jest projekt Nuba, którego głównym celem jest stworzenie sfederowanej platformy IaaS Cloud ułatwiającej łatwe i automatyczne wdrażanie internetowych usług biznesowych, umożliwiającej dynamiczne skalowanie w oparciu o kryteria wydajności i celów biznesowych. Niektóre grupy standaryzacyjne Cloud rozpoczęły prace nad zdefiniowaniem wspólnych interfejsów do współdziałania. Grupa robocza Open Grid Forum Open Cloud Computing Interface (OCCI) przy OGF pracuje nad zdefiniowaniem specyfikacji API do zdalnego zarządzania infrastrukturą chmury



obliczeniowej, co pozwoli na rozwój interoperacyjnych narzędzi do typowych zadań, w tym wdrażania, autonomicznego skalowania i monitorowania. Projekty OpenNebula i RESERVOIR dostarczyły implementacje zgodne z OCCI. Cloud Computing Interoperability Forum (CCIF) to niezależna od dostawcy, nienastawiona na zysk społeczność zwolenników technologii i konsumentów, których celem jest szybkie wprowadzenie globalnych usług przetwarzania w chmurze. Chociaż pojawiają się zachęcające do działań w obszarze chmur interoperacyjnych, technologie gridowe są bardziej dojrzałe. Dlatego obiecujące jest rozszerzenie ich na chmurę, szczególnie w zakresie badań i oceny strategii planowania i wyboru zasobów. Podczas gdy przetwarzanie sieciowe koncentruje się na wykorzystaniu, przetwarzanie w chmurze jest bardziej dostosowane do takich czynników, jak QoS, koszt i efektywność energetyczna. Wreszcie, Chmury będą chciały skorzystać z elastycznego wykorzystania swoich zasobów, aby zoptymalizować zarówno wykorzystanie zasobów (a tym samym przychody dostawców chmury), jak i QoS zapewniane użytkownikom.

### **Bezpieczeństwo i zarządzanie użytkownikami**

Obecnie w chmurach brakuje wielu mechanizmów wymaganych do płynnego działania między lokacjami, których kluczowym czynnikiem jest bezpieczeństwo. Interoperacyjność wymaga wspólnych mechanizmów bezpieczeństwa, które można przełożyć na modele wybrane przez lokalnych administratorów. Dodatkowo użytkownicy muszą mieć możliwość składania wniosków niezależnie od instytucji zaangażowanych w proces wykonywania zleconego zadania lub dostarczania niezbędnych danych. Wymaga to wprowadzenia do Chmury mechanizmu delegowania uprawnień umożliwiającego jednokrotne logowanie. Wreszcie, współpraca między wieloma instytucjami współdzielącymi zasoby wymaga opracowania nowych metod zarządzania uprawnieniami użytkowników. Wyzwania te zostały już rozwiązane w Grid Computing, gdzie głównym celem jest umożliwienie współdzielenia zasobów pomiędzy wirtualnymi organizacjami. VO definiuje grupę osób i zasobów, które mogą pojawiać się w wielu domenach administracyjnych, i umożliwia definiowanie szczegółowych zasad bezpieczeństwa dla tych zasobów. Rozwiązania Grid są opisane w kontekście oprogramowania pośredniczącego Globus i pokazują, w jaki sposób koncepcje można zastosować do Clouds.

Użytkownicy w sieci Grid otrzymują uprawnienia od administratorów witryny na podstawie ich poświadczeń, które są dostarczane przez zaufany urząd certyfikacji. Infrastruktura bezpieczeństwa sieci (GSI) jest składnikiem oprogramowania pośredniczącego Globus odpowiedzialnym za koordynację bezpieczeństwa w różnych lokalizacjach. GSI jest używany przez protokoły wykonywania zadań, przesyłania plików oraz wykrywania i monitorowania zasobów, aby zapewnić, że wszystkie operacje rozpoczęte przez użytkownika są dozwolone w zasobach docelowych. GSI korzysta z infrastruktury klucza publicznego X.509 (PKI) i protokołów SSL/TLS do szyfrowania transportu. Dzięki temu osoby należące do organizacji mogą ufać zagranicznym poświadczeniom wydanym przez urząd certyfikacji bez wpływu na środki bezpieczeństwa ich organizacji. Jednak z dynamicznej natury systemów Grid wynikają dwa dodatkowe wymagania.

Logowanie jednokrotne: użytkownicy w sieci Grid muszą uzyskiwać dostęp do wielu zasobów i usług za pomocą różnych modeli uwierzytelniania. Byłoby uciążliwe, gdyby za każdym razem, gdy użytkownik musiał wykonać akcję w zdalnym zasobie, musiałby wprowadzić hasło, aby użyć swojego klucza prywatnego do uwierzytelnienia się. Możliwe rozwiązania, takie jak buforowanie hasła, mogą prowadzić do dodatkowych problemów z bezpieczeństwem.

Delegowanie uprawnień: ze względu na dynamiczną naturę sieci Grid użytkownicy często muszą delegować swoje uprawnienia do innych usług. Dzieje się tak, gdy żądania wymagają aranżacji różnych zasobów lub gdy użytkownik tworzy nową usługę, aby wykonać zestaw możliwości. Zgodnie z zasadą

najmniejszych uprawnień, zestaw minimalnych możliwości powinien zostać przeniesiony do tych usług, aby mogły one działać.

Wymagania te spełnia rozszerzenie do certyfikatów X.509 zwane certyfikatami proxy. Certyfikaty proxy nie są wydawane przez CA, co byłoby uciążliwe ze względu na częstotliwość ich używania i dynamiczny charakter. Zamiast tego wystawca jest identyfikowany przez inny certyfikat klucza publicznego. Dzięki temu tymczasowe certyfikaty mogą być podpisywane i używane przez krótki czas bez konieczności uzyskiwania dostępu do długoterminowych kluczy prywatnych użytkownika. Zrelaksowane bezpieczeństwo certyfikatów proxy jest wystarczające, ponieważ mają one krótki cykl życia. Certyfikaty proxy są również używane do tworzenia nowych certyfikatów z delegowanymi podzbiorami uprawnień. Architektura GSI pozwala na różne poziomy szczegółowości przy określaniu, które z uprawnień są dziedziczone przez utworzony serwer proxy. Bardziej szczegółowe poziomy szczegółowości można zaimplementować za pomocą języków zasad do wyrażania zasad delegowania. Możliwość ta jest skutecznie wykorzystywana przez bardziej zaawansowane usługi bezpieczeństwa zbudowane na GSI, takie jak usługa CAS opisana poniżej.

Przykładem wykorzystania certyfikatów proxy może być zadanie obliczeniowe, które wymaga dostępu do serwera pamięci masowej w celu ściągania zbiorów danych do przetworzenia. W takim przypadku nowy serwer proxy zostałby utworzony w pierwszej lokalizacji poprzez delegowanie uprawnień użytkownika w sieci, a z kolei zasób odbierający żądanie przekazałby dane uwierzytelniające do serwera magazynu, który wykonałby operację w oparciu o swoje polityki autoryzacji. Jednym z problemów związanych z certyfikatami proxy opartymi na x509 jest konieczność inicjowania przez użytkowników żądań z komputera, na którym przechowywane są ich klucze prywatne, oprócz wymaganego oprogramowania do generowania serwera proxy i uruchamiania żądania do sieci Grid. Często użytkownicy uzyskują dostęp do sieci Grid za pośrednictwem portali internetowych, co utrudnia generowanie ich certyfikatów proxy. Repozytorium poświadczeń MyProxy zostało utworzone w celu rozwiązania tego problemu i umożliwienia każdemu użytkownikowi dostępu do zasobów Grid za pośrednictwem portalu Grid przy użyciu przeglądarki internetowej. Model MyProxy dodaje usługę repozytorium, w której użytkownicy delegują swoje poświadczenia i kojarzą je z nazwą użytkownika i hasłem. Następnie użytkownicy mogą zalogować się do portalu internetowego obsługującego MyProxy i pobrać i wykorzystać wcześniej zapisany certyfikat Grid. Certyfikaty delegowane do repozytoriów MyProxy mają dłuższy czas życia niż zwykłe serwery proxy, więc użytkownicy muszą je od czasu do czasu generować. Infrastruktura GSI umożliwia właścicielom zasobów definiowanie zasad dostępu w sposób adhoc: zwykle administratorzy witryny są odpowiedzialni za zdefiniowanie mapowania z nazw wyróżniających (DN) na lokalną metodę zabezpieczeń. Stwarza to szereg problemów, szczególnie w przypadku dużych VO rozproszonych w różnych instytucjach: pierwszym problemem jest obciążenie administratorów w celu uwzględnienia polityk dostępu dla wszystkich użytkowników, zwłaszcza jeśli istnieje potrzeba zdefiniowania bardziej szczegółowych, które się różnią, z jednego zasobu do drugiego. Po drugie, administratorzy systemów odpowiedzialni za przypisywanie polityk dostępu nie mają dużego obrazu potrzeb projektu w zakresie struktury uprawnień. Community Authorization Service (CAS) (Pearlman, Welch, Foster i Kesselman, 2002) jest rozszerzeniem opartym na GSI, które zapewnia dodatkowe mechanizmy usuwania wyżej wymienionych braków. CAS wyodrębnia złożoność polityk dostępu dla projektu na centralnym serwerze, który działa jako repozytorium polityk i użytkowników, zwalniając lokalnych administratorów zasobów z zadania identyfikacji wymagań autoryzacyjnych. Bezpośrednią korzyścią z takiego oddzielenia obaw jest to, że administratorzy projektu mogą definiować użytkowników i reguły dostępu na serwerze CAS, a nawet tworzyć grupy w celu definiowania szczegółowych zasad. Gdy użytkownicy zostaną dodani do serwera CAS, kontaktują się z nim, gdy potrzebny jest dostęp do zasobu, a serwer CAS nadaje im możliwość równoważną certyfikatowi proxy. Administratorzy witryny muszą jedynie potwierdzić, że zamierzona operacja jest

dozwolona dla społeczności, do której należy użytkownik, oraz że operacja jest dozwolona przez oferowaną funkcję. Ta metoda skaluje się niezależnie od liczby użytkowników i zasobów. Jest bezpośrednio zbudowany na GSI, co pozwala na jego wdrożenie przy minimalnych zmianach w istniejących technologiach. W przypadku przetwarzania w chmurze brak standaryzacji wśród dostawców skutkuje wieloma modelami bezpieczeństwa: na przykład zarówno Amazon EC2, jak i Eucalyptus wykorzystują do uwierzytelniania pary certyfikatów X.509 i kluczy prywatnych. Google App Engine, przykład rozwiązania PaaS, wymaga od użytkowników pierwszego zalogowania się za pomocą kont Google. Różnorodność metod utrudnia tworzenie nowych możliwości współdziałania, a fragmentacja modeli zabezpieczeń utrudnia ponowne wykorzystanie nowo opracowanych funkcji. Grupa Robocza OGF Open Cloud Computing Interface (OCCI) zrobiła krok w kierunku zaproponowania ustandaryzowanego zestawu operacji, a w swojej specyfikacji sugeruje, że implementacje mogą wymagać uwierzytelnienia za pomocą standardowych mechanizmów HTTP i/lub szyfrowania przez SSL/TLS. Najnowsze wersje OpenNebula obsługują tę specyfikację do komunikacji z kontrolerem chmury. Definicja ta reprezentuje możliwość stworzenia wspólnych podstaw dla implementacji IaaS, zapewniając jednolite paradygmaty bezpieczeństwa wśród różnych dostawców. Jednak wciąż jest dużo pracy, aby osiągnąć dobrą infrastrukturę bezpieczeństwa w Chmurze. Metody określania zaufania między wystawcami certyfikatów i właścicielami zasobów nie zostały jeszcze zaimplementowane, zwłaszcza w scenariuszach, w których różne organizacje uczestniczą w ich udostępnianiu. Modele takie jak infrastruktura GSI, w których różni dostawcy ufają różnym Urzędem Certyfikującym bez narażania reszty instytucji, umożliwiłyby skalowanie chmur poza granice jednej instytucji. W takich przypadkach konieczne byłyby dodatkowe techniki zarządzania użytkownikami i powiązaniymi z nimi uprawnieniami, aby uniknąć centralizacji, a także potrzebne byłyby nowe rozproszone metody rozliczania. Chmury mogą uczyć się z tych rozwiązań w celu definiowania nowych, ustandaryzowanych interfejsów, które umożliwiają bezpieczną komunikację między organizacjami.

### **Modelowanie i symulacja chmur i siatek**

Ponieważ ocena różnych zastosowań na rzeczywistych stanowiskach testowych Grid jest trudna lub nawet niewykonalna, opracowano różne symulatory w celu badania złożonych scenariuszy. Symulacje pozwalają nam na badanie polityk dla dużych i złożonych konfiguracji z licznymi zadaniami i wysokim zapotrzebowaniem na zasoby oraz na łatwe włączanie modyfikacji i udoskonalień do polityk. Istnieje wiele bogatych modeli symulacyjnych opracowanych przez społeczność Grid. Symulator GridSim jest szeroko stosowany przez wielu badaczy do oceny strategii planowania sieci. Zgodnie z opisem zespołu projektowego GridSim zapewnia wszechstronne narzędzie do tworzenia różnych klas heterogenicznych zasobów, które można agregować za pomocą brokerów zasobów. GangSim umożliwia symulację złożonych obciążeń i charakterystyk systemu. Jest również w stanie wspierać badania nad kontrolowanym udostępnianiem zasobów w oparciu o umowy SLA. Zestaw narzędzi SimGrid to symulator nieoparty na obciążeniu, który umożliwia ocenę aplikacji rozproszonych w heterogenicznych środowiskach rozproszonych. W tych ostatnich modelach prawie wszystkie modelują pracę zaplanowaną na poziomie wielu lokalizacji (przez danego brokera lub metaplanera), ale nie w jaki sposób zadania są planowane i przydzielane po wysłaniu do końcowych zasobów obliczeniowych. W innym podejściu symulator Alvio i Teikoku modelują wszystkie warstwy planowania, które są zaangażowane w architektury Grid, od zasad meta-brokeringu po lokalne strategie planowania zadań. DGSim to kolejna istotna platforma symulacyjna, która umożliwia również symulację sieci z wykorzystaniem metod meta-brokeringu, ale tak jak poprzednie podejście, nie modeluje lokalne scenariusze. Narzędzia symulacyjne są szczególnie ważne w badaniach nad przetwarzaniem w chmurze ze względu na fakt, że wiele chmur jest nadal w fazie rozwoju. CloudSim modeluje i symuluje środowiska przetwarzania w chmurze obsługujące wiele maszyn wirtualnych w węzle centrum danych. W rzeczywistości zarządzanie maszynami wirtualnymi jest główną nowością

tego symulatora. Umożliwia również symulację wielu sfederowanych centrów danych, aby umożliwić badanie zasad migracji maszyn wirtualnych pod kątem niezawodności i automatycznego skalowania aplikacji. Jednak kilka aspektów przetwarzania w chmurze nie zostało jeszcze rozwiązanych, takich jak symulacja wielu warstw jednocześnie. Dlatego brak symulatorów do przetwarzania w chmurze motywuje do rozbudowy istniejących symulatorów, które zostały opracowane dla systemów Grid i mają podobne wymagania. Niektóre z istniejących symulatorów sieci zostały opisane poniżej. Chociaż wiele modeli symulacji chmury nie zostało jeszcze opracowanych, wykorzystuje się niektóre z tych modeli symulacyjnych i doświadczenia prawdopodobnie przyspieszyłyby rozwój chmury. Obciążenia mają kluczowe znaczenie dla oceny zasad przy użyciu narzędzi symulacyjnych. Chociaż zaproponowano różne modele obciążenia, ślady z dzienników systemów produkcyjnych lepiej oddają zachowanie realistycznych scenariuszy. Istnieją różne publicznie dostępne ślady obciążenia z systemu produkcyjnego, takie jak te dostarczane przez Obserwatorium sieci, które zbiera, publikuje i analizuje dane dotyczące zachowania sieci EGEE. Jest to obecnie jeden z najbardziej złożonych publicznych śladów Grid z wyższą częstotliwością nadejścia żądań aplikacji niż inne duże Grids, takie jak Grid5000. Jednak każda z nich oddaje heterogeniczny charakter zwirtualizowanych infrastruktur chmurowych z wieloma rozproszonymi geograficznie punktami wejścia i potencjalnie wysokimi wskaźnikami przybycia miejsc pracy. Ponadto, ponieważ ślady z różnych systemów są w różnych formatach, korzystanie ze standardowych formatów jest bardzo ważne. W ramach ParallelWorkload Archive, a także dostarczanie szczegółowych dzienników obciążeń zebranych z dużych systemów równoległych w zastosowaniach produkcyjnych, takich jak San Diego Supercomputer Center lub Los Alamos National Lab, Feitelson i in. proponuje Standardowy format obciążenia (SWF), który został zdefiniowany w celu ułatwienia korzystania z dzienników i modeli obciążenia. Iosup i in. rozszerzył ten pomysł na Grids o GridWorkload Archive oraz o Failure Trace Archive, aby ułatwić projektowanie, walidację i porównanie modeli i algorytmów odpornych na błędy. Brakuje śladów obciążeń i standardowych modeli dla Clouds. Jest to ważna przeszkoda w modelowaniu i symulowaniu realistycznych scenariuszy przetwarzania w chmurze, ponieważ obciążenia chmury mogą składać się z różnych typów aplikacji, w tym żądań usług, które mają inne zachowanie niż modelowane w bieżących publicznych śladach. Te istniejące podejścia do systemów równoległych i systemów Grid można rozszerzyć na przetwarzanie w chmurze z definicją standardowego formatu obciążenia chmury. Dzienniki obciążenia zebrane z produkcji lub badań Systemy chmurowe powinny być również udostępniane publicznie, aby ułatwić badanie technik przetwarzania w chmurze poprzez symulację.

### **Uwagi końcowe**

Siatki i chmury mają wiele podobieństw w architekturze, technologiach i technikach. Obecnie wydaje się, że przetwarzanie w chmurze nabiera coraz większego znaczenia jako sposób oferowania elastycznej platformy dostępu do zdalnych zasobów przetwarzania: jest to poparte rosnącym zainteresowaniem rynku nowymi platformami, liczbą nowych firm, które korzystają i dostarczają usługi w chmurze oraz zainteresowaniem środowiska akademickiego tym nowym paradygmatem. Jednak nadal istnieje wiele aspektów przetwarzania w chmurze, którymi należy się zająć, takich jak uzależnienie od dostawcy, obawy dotyczące bezpieczeństwa, lepsze systemy monitorowania itp. Wierzymy, że technologie opracowane w Grid Computing można wykorzystać do przyspieszenia dojrzałości Chmura i nowe możliwości oferowane przez tę drugą w perspektywie czasowej zajmą się niektórymi niedociągnięciami sieci Grid. Jak stara się przekazać ta część, być może obszarem, w którym Clouds może najwięcej zyskać na technologiach gridowych, jest interoperacyjność wielu lokalizacji. Wynika to naturalnie z faktu, że głównym celem systemów Grid jest umożliwienie zdalnym witrynom, w ramach różnych zasad administracyjnych, ustanowienia wydajnej i zorganizowanej współpracy. Jest to prawdopodobnie jeden z najsłabszych punktów w chmurach, które zwykle są usługami oferowanymi przez pojedyncze organizacje, które wymuszają swoje - często zastrzeżone - protokoły, co prowadzi do

przykładów już zidentyfikowanego problemu blokady dostawcy. Z drugiej strony, przetwarzanie siatkowe, dzięki zastosowaniu dobrze zdefiniowanych standardów, osiągnęło interoperacyjność w miejscu, o czym świadczy wiele siatek obliczeniowych i siatek danych wykorzystywanych w projektach w dziedzinach takich jak fizyka cząstek elementarnych, nauki o Ziemi, genetyka i nauki ekonomiczne. Inną ścieżką, w którą warto się zagłębić, jest ta badająca, w jaki sposób nowy paradygmat przetwarzania w chmurze może przynieść korzyści istniejącym technologiom i rozwiązaniom proponowanym przez społeczność Grid: realizacja obliczeń użytkowych, elastyczne dostarczanie zasobów lub homogenizacja heterogenicznych zasobów (pod względem sprzętowym, systemy operacyjne i biblioteki oprogramowania) poprzez wirtualizację otwierają nową sferę możliwych zastosowań ogromnych, niewykorzystanych zasobów obliczeniowych. Nowe techniki konsolidacji pozwalają na badania nad niższym zużyciem energii przez centra danych i obniżonymi kosztami dla użytkowników zasobów obliczeniowych. W rzeczywistości istnieje nowa gama aplikacji, które można uruchamiać w chmurach ze względu na lepszą izolację zapewnianą przez techniki wirtualizacji. W ten sposób istniejące oprogramowanie, które było trudne do uruchomienia w sieciach Grid ze względu na twarde zależności od bibliotek i/lub systemów operacyjnych, może być teraz wykonywane na znacznie większej liczbie zasobów, które zostały aprowizowane zgodnie z wymaganym środowiskiem. Na koniec należy wziąć pod uwagę kilka nierozstrzygniętych problemów, które uniemożliwiają niektórym użytkownikom przejście na nowe technologie w chmurze. Te problemy należy rozwiązać, zanim będziemy mogli w pełni wykorzystać wszystkie wymienione możliwości. Inni autorzy, tacy jak , wymienili już kilka takich problemów. Oto kilka przykładów:

1. W niektórych przypadkach, gdy procesy wymagają intensywnego wykorzystania I/O, środowiska zwirtualizowane oferują niższą wydajność niż zasoby natywne. Istnieje szereg aplikacji naukowych, które mają duże zapotrzebowanie na komunikację, na przykład te, które opierają się na synchronicznych modelach przekazywania wiadomości. Te aplikacje nie oferują dobrej wydajności w systemach Cloud.

2. Mimo że Chmury oferują obietnicę elastyczności zasobów obliczeniowych, która jawi się użytkownikom jako nieskończona podaż, w świecie naukowym zdarzają się scenariusze, dla których zasoby oferowane przez pojedynczą chmurę nie wystarczą. Gdy zapotrzebowanie na moc obliczeniową osiągnie maksymalną wydajność dla dostawcy, nie ma dodatkowych środków na pozyskanie nowych zasobów dla użytkowników, jeśli zajdzie taka potrzeba. Próba użycia różnych dostawców jako kopii zapasowej oznaczałaby zastosowanie różnych protokołów, schematów bezpieczeństwa i nowych interfejsów API. Na przykład projekt Wielkiego Zderzacza Hadronów (LHC) wymaga mocy obliczeniowej niedostępnej dla żadnej pojedynczej organizacji, a jeśli zostanie wdrożony w chmurze, istnieje potrzeba interoperacyjności między różnymi dostawcami chmury.

Mamy nadzieję, że wysiłki podejmowane przez wielu badaczy w tej dziedzinie zidentyfikują i usuną te niedociągnięcia oraz doprowadzą do stworzenia lepszych i bardziej dojrzałych technologii, które poprawią obecne praktyki przetwarzania w chmurze. W ramach tych wysiłków wierzymy, że dobra znajomość istniejących technologii, technik i architektur, takich jak te opracowane w dziedzinie Grid Computing, znacznie pomoże przyspieszyć tempo badań i rozwoju chmury oraz zapewni lepsze przejście do tego nowego paradygmaty komputerowe.