Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Triennale in Informatica

POSSIBILI SOLUZIONI PER GARANTIRE QoS NELLE COMUNICAZIONI INTER-DATA CENTER IN AMBIENTI CLOUD COMPUTING

Tesi di Laurea in RETI DI CALCOLATORI

Relatore:
Chiar.mo Prof.
FABIO PANZIERI

Presentata da:
PIETRO PELOSO

Sessione II
Anno Accademico 2012-2013

Ai miei genitori che mi hanno dato l'opportunità di raggiungere questo traguardo.

A Mariangela che più di tutti mi ha aiutato nella stesura della tesi con suggerimenti, critiche e osservazioni.

A tutti i miei amici e coinquilini, conosciuti durante questi anni, che mi hanno sostenuto e aiutato a crescere durante questo percorso.

Introduzione

L’espressione cloud computing fa riferimento a un nuovo paradigma per la fornitura di infrastrutture informatiche, il quale prevede che tali infrastrutture siano localizzate nella rete al fine di ridurre i costi di gestione delle risorse hardware e software. Come si avrà modo di vedere, si tratta dell’evoluzione di una serie di tecnologie prima utilizzate individualmente, che adesso invece, utilizzate congiuntamente, stanno rivoluzionando le modalità con cui le organizzazioni costruiscono le proprie infrastrutture informatiche.

L’elasticità e il risparmio economico sono alcuni dei vantaggi del cloud computing in quanto l’infrastruttura che solitamente ospita i servizi cloud può avere diversi livelli di virtualizzazione e i suoi costi possono essere ottimizzati con la multi-tenancy; grazie a quest’ultima, il cloud viene utilizzato da più tenant con dei meccanismi di protezione e isolamento che tutelano ogni utilizzatore dagli altri.

Seguendo un ordine crescente di complessità, i più popolari modelli di servizio per il cloud computing, di cui si parlerà specificamente nel primo capitolo, sono tre:

1. *Infrastructure as a Service (IaaS)*;
2. *Platform as a Service (PaaS)*;
3. *Software as a Service (SaaS)*.

Ad ogni modo, una delle problematiche più ostiche relative al cloud computing riguarda la performance della connessione tra i data centers e il resto del mondo. Per tale motivo nel corso del lavoro si

analizzeranno alcune delle soluzioni volte a salvaguardare la QoS (Quality of Service) nell'ambiente cloud, in particolare nelle architetture IaaS. Le diverse tipologie saranno prima analizzate singolarmente, basandosi sugli studi presenti in letteratura, e poi confrontate tra di loro sulla base dei seguenti fattori:

- Scalabilità, efficienza e predicibilità;
- Attenzione alla tutela dei singoli tenant e predicibilità del prezzo;
- Equilibrio tra espressività e complessità;
- Bilanciamento tra disponibilità, performance e costi;
- La garanzia deve tenere conto del fatto che un'applicazione non è sempre utilizzata in modo uniforme nello spazio e nel tempo.

Scopo della presente disamina è sostanzialmente studiare le metodologie utilizzate per garantire la QoS nelle comunicazioni fra data-centers in ambiti di cloud computing. Essa si suddivide in tre capitoli.

Il primo di questi tratterà del cloud computing, mostrando una panoramica generale e le caratteristiche principali, per poi passare ai modelli di distribuzione e ai modelli di servizio, suddivisi appunto in IaaS (sezione 1.3.1), PaaS (sezione 1.3.2) e SaaS (sezione 1.3.3).

Continuando, nel secondo capitolo si esamineranno le Netowork Performance nel cloud, offrendo anche in tal caso una panoramica di ciò che si andrà a considerare, soffermando l'attenzione sui diversi sistemi atti a garantire un adeguato QoS, vale a dire DRL, SecondNet, Seawall, Topology Switching, Gatekeeper, Oktopus, FairCloud, Location Indipendence, Choreo, Proteus e Griphon.

Nel terzo capitolo, si procederà con un confronto tra le varie soluzioni proposte in relazione ad alcuni parametri critici, per illustrare, nel capitolo conclusivo, i possibili sviluppi futuri.

Indice

Capitolo 1
Cloud Computing

1.1 Panoramica

Il NIST[1], ovvero l'istituto per la standardizzazione del settore delle tecnologie, ha definito il cloud computing nel modo seguente [1]:

> *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*

Sotto una certa prospettiva il cloud computing non è nulla di nuovo, poiché in merito non troviamo nozioni, criteri e approcci che già non conosciamo. Lo consideriamo nuovo perché ha cambiato il modo di inventare, distribuire, gestire e pagare infrastrutture su cui vengono utilizzate le nostre applicazioni. È un nuovo modo di concepire l'erogazione di servizi Internet-based e la relativa gestione delle risorse come server condivisi, software e dati su

[1] National Istitute of Standars and Technology, agenzia del governo degli Stati Uniti che si occupa della gestione delle tecnologie.

computer e altri dispositivi, richiedendo agli utenti il minimo sforzo per la gestione dell'infrastruttura. L'insieme delle tecnologie informatiche prese in considerazione consentono l'utilizzo delle risorse e dei servizi in remoto, forniti on-demand.

Nel cloud computing vengono offerti come servizi l'utilizzo di software e hardware, in modo tale che l'utente possa accedervi ed usufruirne senza possedere le conoscenze e le modalità operative. Questo modello è basato su di un servizio pay-per-use, ovvero l'utente paga solo l'effettivo utilizzo dei servizi e delle risorse messe a disposizione dal fornitore. Il modello permette quindi di utilizzare le sue funzionalità da un qualsiasi dispositivo che possa connettersi a internet.

1.2 Caratteristiche principali

Prendendo come riferimento la sopracitata definizione del NIST [1], il cloud computing possiede le seguenti caratteristiche:

- ***On-demand self-service.*** Gli utenti possono ottenere le risorse e i servizi richiesti, come capacità di calcolo o archiviazione, ogni qualvolta ne hanno bisogno e senza la necessità di interagire con ogni provider del servizio. Il servizio offerto è completamente automatizzato e il cliente può modificarlo in qualsiasi momento secondo una logica di pay-per-use.
- ***Accesso tramite la rete.*** Le risorse sono accessibili da qualunque punto della rete internet, attraverso protocolli e maccanismi standard che promuovono l’uso di piattaforme client eterogenee (ad esempio computer, tablet, smartphone, etc.).

- ***Raggruppamento delle risorse.*** Le capacità computazionali del provider sono raggruppate per servire più consumatori utilizzando il modello multi-tenant[2], con molteplici risorse fisiche e virtuali assegnate e riassegnate dinamicamente in base alle loro richieste. Di solito i clienti non sanno e non hanno il controllo sulla locazione fisica delle risorse messe a disposizione (RAM, CPU, disco), tuttavia il provider può lasciare che il cliente ne specifichi il luogo in modo astratto, come città, stato o data center. Infatti, non conoscere l'esatta posizione delle risorse può, in alcuni casi, essere un fattore negativo, ad esempio sul decidere quale legge applicare in caso di reato (se i dati risiedono in un altro stato, si applicherà la legge dell'altro stato).
- ***Elasticità.*** Le risorse possono essere fornite e rilasciate, in alcuni casi automaticamente, per permettere di adeguare rapidamente i mezzi impegnati in base alla domanda. Per il consumatore le risorse disponibili sembrano quasi illimitate e possono essere acquisite in qualsiasi quantità e momento. La possibilità di allocare e rilasciare le risorse presenta un grande vantaggio per i clienti, poiché non è necessario premunirsi di enormi quantità di risorse, ma allocarle in base all'effettiva necessità in tempi molto brevi, praticamente istantanei. Tale modalità di allocazione delle risorse permette di ottenere un risparmio significativo.
- ***Servizio misurato.*** I sistemi cloud controllano e ottimizzano automaticamente l'uso delle risorse, sfruttando un sistema di monitoraggio e accounting ad un livello di astrazione appropriato al tipo di servizio (utilizzo di spazio di archiviazione, di CPU o di banda). L'utilizzo delle risorse può essere monitorato e controllato offrendo una

[2] Letteralmente multi-cliente, detto anche "one to many".

garanzia di trasparenza sia nei confronti del provider, sia dei clienti che usufruiscono del servizio.

1.2 Modelli di distribuzione

A seconda della locazione fisica dell'infrastruttura, attraverso cui è offerto il servizio, possiamo distinguere quattro tipologie di cloud [1]:

- ***Private cloud.*** L'infrastruttura è pensata per essere utilizzata da una singola organizzazione che comprende molteplici utenti (ad esempio più unità commerciali). Essa può essere posseduta, mantenuta e gestita dall'organizzazione in questione, da una terza parte o da una combinazione delle due.
- ***Community cloud.*** In questo modello l'infrastruttura è pensata per l'uso esclusivo di una comunità di consumatori, oppure da parte di più organizzazioni che fanno parte della stessa comunità. Può essere di proprietà, mantenuta e gestita da una o più organizzazioni della comunità, da un terzo, o da una combinazione di essi.
- ***Public cloud.*** In questo modello l'infrastruttura è fornita per un pubblico generico. Può essere gestita, posseduta e mantenuta da un'azienda, da un'organizzazione accademica o governativa, da un parte terza o una combinazione di queste.
- ***Hybrid cloud.*** In questo modello l'infrastruttura è una composizione di due o più cloud distinti (private, community o public) che rimangono delle entità separate, ma legate tra loro da una tecnologia condivisa che rende possibile la condivisione e la portabilità dei dati e delle applicazioni.

1.3 Modelli di servizio

I modelli di servizio del cloud computing possono essere rappresentati sotto forma di piramide, man mano che ci si avvicina alla base aumenta l'autonomia per lo sviluppo e la personalizzazione da parte dei clienti, ovviamente ad un prezzo maggiorato.

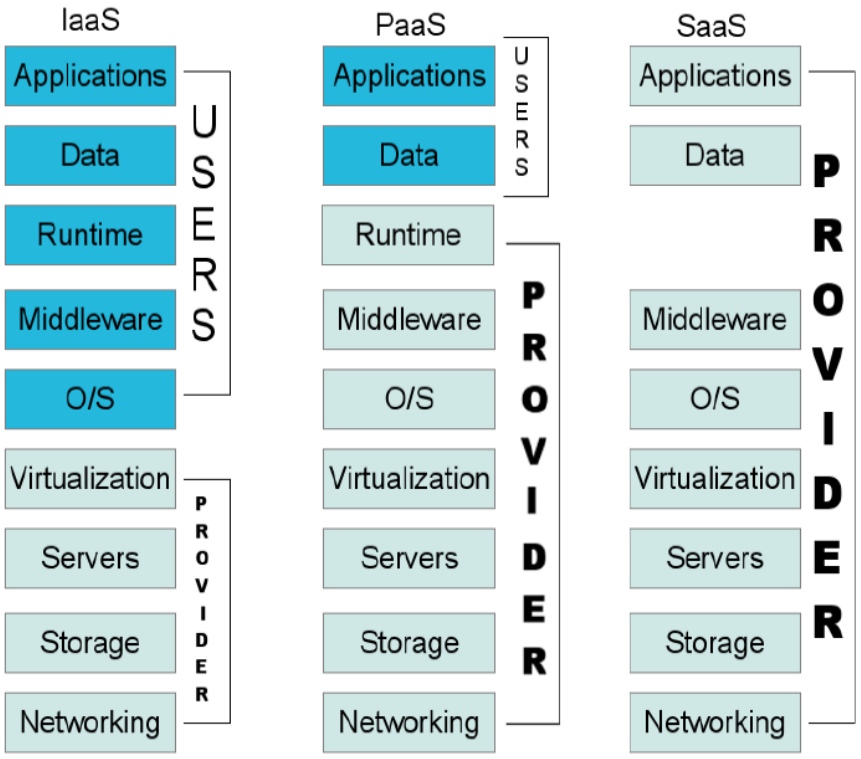

Figura 1. Separazione delle responsabilità nel cloud computing.

I modelli trattati sono [1]:

- *Infrastructure as a Service* – Iaas;
- *Platform as a Service* – PaaS;
- *Software as a Service* – SaaS.

Queste tre tipologie sono collegate tra loro, ovvero possiedono come modello base lo IaaS, ma caratterizzati da livelli di astrazione e complessità differenti.

1.3.1 Infrastructure as a Service – Iaas

L'Infrastructure as a Service dà la possibilità al cliente di affittare capacità di CPU, network, storage e altre risorse che l'utente è in grado di implementare e gestire. Il cliente ha il controllo sui sistemi operativi e su eventuali risorse aggiuntive e potrebbe, inoltre, avere un controllo limitato sui componenti di rete (come ad esempio il firewall). E' da ricordare che l'utente non ha il controllo sull'infrastruttura di base del cloud.

Lo Iaas crea una versione virtuale di una risorsa normalmente fornita fisicamente. L'acquisto, la gestione e la manutenzione delle risorse hardware resta sempre compito del provider, l'utente si occuperà dell'acquisto della CPU, della larghezza di banda, della memoria, necessarie per l'esecuzione e lo sviluppo delle proprie applicazioni.

Il vantaggio di questa soluzione è appunto la capacità di controllo consentita all'utente. Scelto l'hardware più opportuno con il contratto di fornitura del servizio è poi possibile sfruttare da remoto, da computer poco potenti o da cellulari, la potenza presente su altri computer, ed eseguire elaborazioni complesse e che possono richiedere molto tempo.

Uno dei migliori esempi di cloud computing IaaS è la piattaforma di Amazon EC2. Amazon ha creato, come precursore nel 2006, un modello di cloud aderente ai vantaggi sopra citati. I sevizi di Amazon consentono di creare cloud server scalabili, con valori differenti di CPU e RAM, pagando esclusivamente le singole ore di computazione. A completamento del servizio vi sono in aggiunta servizi di storage e di networking che consentono ai clienti di creare, all'interno di EC2 una vera e propria infrastruttura di data center virtuale.

Amazon dà la possibilità di operare facilmente con una infrastruttura virtuale scalabile sia a piccole start-up e privati sia a grandi multinazionali: il numero di istanze richieste nella cloud non rappresenta mai un problema, Amazon è infatti in grado di aggiungere rapidamente hardware nei propri data center. Questo ci porta ad un altro aspetto fondamentale: Amazon gestisce diversi data center per il proprio cloud, il che assicura al cliente anche una ridondanza di tipo geografico per il servizio.

1.3.2 Platform as a Service – PaaS

Il servizio PaaS consiste nella virtualizzazione di una piattaforma completa nella quale si svilupperanno applicazioni. L'utente non deve occuparsi di come è stata creata l'infrastruttura dove si realizzerà la piattaforma, poiché questo è compito del provider del servizio PaaS, che fornirà l'hardware, il sistema operativo e le relative librerie. All'utente spetteranno solo i compiti di creazione e gestione delle applicazioni. Tipicamente queste piattaforme rendono disponibili strumenti di sviluppo, di creazione di interfacce web e servizi web, storage e database.

Il vantaggio di questa soluzione è la possibilità di semplificare il processo di sviluppo delle applicazioni; la procedura esula però dai problemi di costo e complessità legati all'acquisto e alla gestione dello strato hardware/software sottostante. Inoltre, in una soluzione PaaS, il provider fornisce all'utente anche i tool necessari per agevolare il supporto al ciclo di vita del software, dallo sviluppo al rilascio.

1.3.3 Software as a Service – SaaS

Software as a Service è un servizio software che poggia su una piattaforma, poggiata a sua volta su un'infrastruttura. Di norma è un software completo,

creato su misura per il cliente, e spesso può essere personalizzabile e configurabile. Il software ha una completa trasparenza per il cliente, informandolo su dove è ospitato, in che linguaggio di programmazione è stato scritto, su che sistema operativo è installato. È il servizio di più alto livello che possa essere offerto. L'accesso e l'utilizzo del software può essere effettuato da qualsiasi dispositivo dotato di browser web.

Capitolo 2

Network Performance nel Cloud

2.1 Panoramica

Senza reti ad alte prestazioni non avremmo il cloud computing [3]. I data center del cloud vi dipendono per collegare i loro server con il resto del mondo.
I provider del servizio IaaS offrono diversi livelli di servizi e prezzi a seconda delle diverse dimensioni delle macchine virtuali, della memoria, dello storage, etc. Raramente vengono offerte promesse circa le prestazioni della rete - larghezza di banda, perdita di dati, latenza. Così facendo, i clienti potrebbero arrivare a pagare le conseguenze delle prestazioni di una rete imprevedibile.
Molti clienti vorrebbero poter contare su delle garanzie per quanto riguarda le prestazioni della rete e anche molti provider vorrebbero offrire, a pagamento, queste garanzie, ma la maggior parte di loro non le fornisce.
Alcuni studi sulle prestazioni delle applicazioni [4] hanno dimostrato che i cloud network senza garanzie soffrono di latenze alte e molto variabili e di alti tassi di perdita. Finora nessuno ha ancora stabilito quali sono le garanzie per definire una "network performance" in modo che soddisfi i clienti e che sia implementabile dai providers.
Esaminiamo alcune proposte che provvedono a garantire le prestazioni all'interno dei cloud networks.

2.2 Distributed Rate Limiting

Raghavan et al. [5] si concentrano su un aspetto particolare, ovvero il controllo della larghezza di banda globale utilizzata da un servizio cloud, chiamato Distributed Rate Limiting (DRL). L'obbiettivo è lasciare che un insieme di limitatori di velocità del traffico della rete collaborino, per sottoporre una determinata categoria di traffico (ad esempio, il traffico di un particolare servizio basato sul cloud), ad un unico limite globale. Ad esempio, un provider con 10 hosting center può decidere di diminuire la quantità totale del traffico per un particolare servizio a 100 Mbps. Le due opzioni saranno: limitare il servizio a 100 Mbps per ogni hosting center (con il rischio che tutte utilizzano questo limite contemporaneamente) oppure ogni centro con una quota fissa (ad esempio 10 Mbps) limitando il traffico di un servizio. E' improbabile che esso permetta al servizio di consumare tutto il budget assegnato, a meno che il traffico non sia perfettamente bilanciato.

La sfida del DRL è di permettere ai singoli flussi di contendersi in modo dinamico la larghezza di banda, non solo con i flussi che attraversano lo stesso limitatore, ma anche con quei flussi che attraversano diversi limitatori. Pertanto i flussi in arrivo a diversi limitatori dovrebbero raggiungere le stesse dimensioni, esattamente come se attraversassero un unico limitatore condiviso. L'imparzialità dei flussi all'interno del traffico globale dipende strettamente dai compiti del limitatore, che a sua volta dipende dal tasso di arrivo dei pacchetti locali, dal numero di flussi, e dalla capacità up/down-stream[3] del bottleneck. Raghaven et al. [5] questo tema presentando l'illusione di passare tutto il

[3] Upstream: velocità di trasferimento dei dati dal client verso il server, Downstream: velocità di trasferimento dei dati dal server verso il client.

traffico tramite un unico limitatore token-bucket[4], permettendo ai flussi di competere uno contro l'altro per la larghezza di banda secondo le modalità previste nel protocollo di trasporto in uso. La chiave per fornire questa astrazione è misurare la domanda dell'ammontare complessivo ad ogni limitatore, e distribuire la capacità in proporziona alla richiesta.

2.3 SecondNet

SecondNet è un'architettura virtualizzata di data center [6] [Figura 2]. Essa si concentra sull'allocazione della bandwidth e sfrutta i server hypervisor technology per la CPU, memoria e storage isolation e sharing.

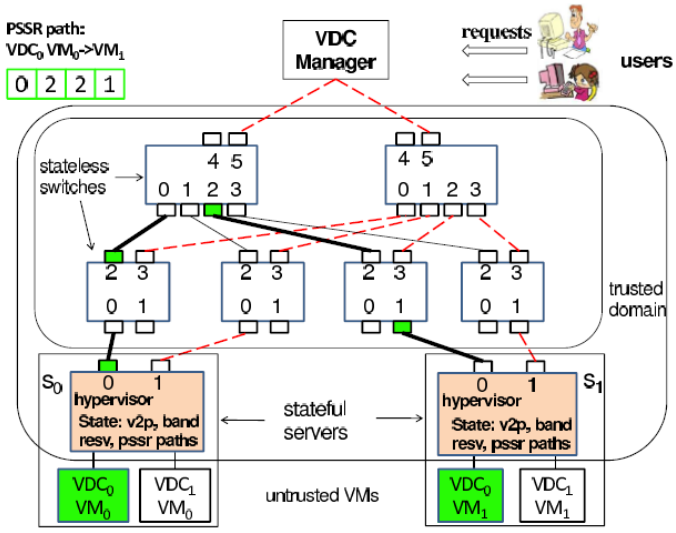

Figura 2. L'architettura SecondNet. Le linee tratteggiate rosse formano uno spanning tree di segnalazione. Le linee nere indicano un percorso port-switching source routing (PSSR). [6]

Un Virtual Data Center (VDC) è definito come un insieme di macchine virtuali con un range di indirizzi IP forniti dal cliente e di un relativo accordo sul livello di servizio (SLA). Lo SLA non specifica solo la computazione e i requisiti di

[4] Il token bucket è un meccanismo di controllo di trasmissione che determina quando e quanto traffico dati (sotto forma di pacchetti) può essere trasmesso in base alla presenza o meno di token(gettoni) in un contenitore astratto, che detiene il traffico complessivo di rete da trasmettere, detto appunto bucket.

archiviazione, ma anche requisiti della larghezza di banda per le macchine virtuali. Un VDC risulta più desiderabile per un cliente, rispetto a un data center fisico grazie all'elasticità che permette di regolare lo SLA in base alle sue richieste. Gli obbiettivi di SecondNet sono i seguenti:

- Il design deve essere scalabile.
- Si deve raggiungere un elevato utilizzo dell'infrastruttura di rete.
- Elasticità quando le esigenze del cliente cambiano.
- Fornire garanzie per la larghezza di banda.

Per provvedere alle garanzie per la larghezza di banda, sfruttiamo una caratteristica importante dei data centers network. Un network di data center è amministrato da un unico soggetto, in modo tale che la topologia della rete, i guasti interni e la distribuzione centralizzata della larghezza di banda, insieme con la gestione degli errori, possono essere gestiti più facilmente. Aldilà dei vantaggi sopra illustrati, si possono incontrare alcune difficoltà quando le Integrated Service[5] dovranno raggiungere i parametri di QoS richiesti, a causa dei numerosi provider coinvolti.

Anche la distribuzione centralizzata della larghezza di banda pone delle ardue sfide, essendo un problema NP-hard [6]. Per risolvere questo problema è stato sviluppato un algoritmo euristico con bassa complessità d'esecuzione. Raggruppiamo server vicini in cluster di dimensioni diverse. Quando assegniamo un VDC, cerchiamo solo i cluster appropriati anziché tutta la rete fisica, riducendo notevolmente il tempo di assegnamento. Questo porta anche a una larghezza di banda efficiente per i VDC, poiché i server raggruppati risultano abbastanza vicini. In questo modo quindi è stato usato un algoritmo

[5] Indica un modello di servizio integrato (posto a livello 3 della pila protocollare ISO-OSI) che si adatta piuttosto bene a molti degli aspetti richiesti dalla QoS.

efficiente e dal costo esiguo per mappare le VM su server fisici e sfruttare la ricca connettività delle reti fisiche per l'assegnazione dei percorsi.

Per un implementazione pratica introduciamo il Port-Switching basato su Source Routing[6] (PSSR).

Dato che la topologia di rete del data center network è conosciuta, il PSSR rappresenta un percorso di routing visto come una sequenza di porte di output di switch. Il PSSR può essere implementato utilizzando le funzionalità del MPLS (Multi-Protocol Label Switching)[7].

La simulazione dei risultati del loro algoritmo mostra che si possono allocare 5000 VM nel VDC in 493 secondi di media in data center con 100000 server. Gli esperimenti dimostrano che SecondNet fornisce la differenziazione del servizio e la garanzia di banda, può eseguire un percorso di allocazione in pochi secondi e lo spostamento delle VM in una decina di secondi per la gestione degli errori e dell'espansione dinamica del VDC.

Abbiamo tre modelli di servizi per i VDC:

- Il servizio type-0 fornisce garanzie di bandwidth tra due VM, che è analogo al Integrated Service.
- Il servizio best-effort tradizionale, senza alcuna garanzia di banda.
- Tra questi due servizi ne abbiamo un terzo, type-1che dà la possibilità di riservare localmente la bandwidth d'entrata e di uscita per la VM.

Il modello di VDC preso in considerazione si concentra sulla larghezza di banda, in quanto risulta essere una risorsa limitata. Type-0 avrà la più alta

[6] Il source routing è una tecnica in cui un nodo, originatore di un pacchetto, inserisce all'interno del pacchetto stesso il percorso che esso dovrà effettuare.

[7] E' una tecnologia per reti IP che permette di instradare flussi di traffico multiprotocollo tra nodo di origine (Ingress Node) e nodo di destinazione (Egress Node) tramite l'utilizzo di identificativi (label) tra coppie di router adiacenti e semplici operazioni sulle etichette stesse.

priorità nel traffico, seguita da type-1 e in seguito dal best-effort, che avrà la più bassa priorità.

2.4 Seawall

Shieh propone Seawall [7], che "*ottiene la massima equità tra le macchine virtuali dei clienti, generando il traffico attraverso una congestione controllata, tramite un tunnel hypervisor-to-hypervisor*". Seawall fa affidamento sull'esecuzione di un controller negli end-host, per fornire un meccanismo di supporto scalabile e il monitoraggio della rete software-based e per fornire inoltre flessibilità e bassi costi nei feedback end-to-end. Seawall posiziona i rate controller nell'hypervisor[8] al fine di proteggerlo dal codice dannoso dell'affittuario, e utilizza i recenti progressi nello sfruttare le schede di rete (NIC) multi-coda e di CPU multicore per raggiungere bassi overhead[9] sugli end-host. Seawall è stata progettata per rimodernare i data center virtualizzati, fornendo performance isolation senza ulteriori preamboli sulle funzionalità del hardware, richiedendo solo un piccolo numero di cambi sul software del end-host e la configurazione dello switch. Infine, Seawall può garantire le performance isolation anche quando gli hypervisor sono compromessi.

Il servizio usa un rate controller basato sul hypervisor, trasmesso dal feedback della rete e dall'hypervisor ricevente, che regola tutto il traffico generato

[8] Hypervisor: è il livello software che si interpone tra il sistema operativo installato nella macchina fisica (host), e quello instanziato nell'ambiente virtuale, definito in genere come guest. L'hypervisor offre ai livelli superiori della macchina virtuale l'interfaccia per dialogare con il sistema operativo ospitante.

[9] Si riferisce a quella parte di banda di trasmissione che viene utilizzata per spedire, anziché l'informazione utile, dati aggiuntivi necessari per i protocolli di trasmissione stessa e per il monitoraggio, la gestione e il controllo della rete stessa, sia da parte di meccanismi automatici che da parte di sistemi di gestione esterni.

dall'affittuario. Così facendo, Seawall è in grado di controllare anche i clienti che generano traffico UDP o che usano in modo anomalo lo stack TCP. Gli affittuari malintenzionati non possono attaccare il rate controller direttamente con lo spoofing[10] come non possono sfuggire al rate controller senza interrompere il hypervisor isolation. Il rate controller protegge anche dagli attacchi denial of service[11]. Seawall utilizza Layer 3 (IP) per segnalare i feedback, il quale può attraversare arbitrarie topologie di data center. Si assume che il gateway Internet del data center partecipi in Seawall come qualsiasi altro compute node.

I rate controller di Seawall sono implementati in una scheda di rete virtuale, ovvero il componente hypervisor responsabile di esportare un'interfaccia del network device a un driver di rete ospite. Il rate controller prende come input i pacchetti ricevuti e inviati al compute node e al congestion feedback dalla rete e dal ricevente. Sul percorso di ricezione, i controlli della scheda di rete virtuale per i segnali di congestione, come i pacchetti persi, inviano questo tipo di feedback al mittente. Sul percorso d'invio la scheda di rete virtuale classifica i pacchetti in ingresso in code per-(sourceVM, destinationVM, percorso), con le destinazioni esterne mappate sul gateway Internet. Il percorso è necessario per le reti che utilizzano un multipath (ECMP[8]) per assegnare i pacchetti con lo stesso protocollo TCP/UDP 5-tuple a percorsi diversi.

Seawall si concentra su un scala di un gran numero di affittuari e macchine virtuali, e su un uso efficiente della larghezza di banda (evitando riserve di risorse fisse). Tuttavia, il sistema non fornisce la prevedibilità delle prestazioni.

[10] Lo spoofing è un tipo di attacco informatico dove viene impiegata in qualche maniera la falsificazione dell'identità (spoof).

[11] Nella sicurezza informatica DoS è la sigla di denial of service, letteralmente negazione del servizio. Si tratta di un malfunzionamento dovuto ad un attacco informatico in cui si esauriscono deliberatamente le risorse di un sistema informatico che fornisce un servizio, ad esempio un sito web, fino a renderlo non più in grado di erogare il servizio.

2.5 Topology Switching

Webb ha osservato che diverse applicazioni hanno differenti requisiti per le performance della rete. Il suo progetto lascia alle applicazioni le specifiche richieste sulla topologia fisica che sono alla base delle loro reti virtuali [9]. Topology Switching (TS) è un modo totalmente diverso di far interagire le applicazioni dei data center con la rete.

Una rete TS supporta multiple e simultanee routing task dalle specifiche applicazioni, come le VLAN[12]. A differenza di quest'ultima, all'interno di ogni attività di routing, l'applicazione può definire diverse topologie, naming e convenzioni di routing studiate appositamente per la loro affidabilità, performance e scalabilità richiesta, garantendo migliori prestazioni a scapito di determinate qualità di routing che possono non essere fondamentali per la routing task in questione. TS tenta di affrontare le sfide poste dai due principali processi di gestione delle risorse delle reti dei data center, ossia il collocamento delle VM e l'evoluzione della rete.

La maggior parte delle proposte di reti dei data center cercano di separare il posizionamento dalle prestazioni, massimizzando la larghezza di banda tra coppie di server, attraverso topologie simmetriche. Mentre questa strategia ammette una notevole flessibilità nell'assegnazione del lavoro in tutto il data center, ci sono casi in cui i modelli di comunicazione sono alterati. Webb sostiene che, per questi carichi di lavoro, è di gran lunga più efficace selezionare i sistemi di routing per applicazioni specifiche, in modo da rendere efficiente l'uso delle risorse fisiche. I gruppi di VM possono infatti ospitare una

[12] In telecomunicazioni e informatica il termine VLAN (Virtual LAN) indica un insieme di tecnologie che permettono di segmentare il dominio di broadcast, che si crea in una rete locale basata su switch, in più reti locali logicamente non comunicanti tra loro, ma che condividono globalmente la stessa infrastruttura fisica di rete locale.

varietà di servizi in cui ognuno di essi ha diverse esigenze di networking. Ad esempio, i servizi ospitati possono essere loro stessi raggruppati a più livelli, oppure replicati.

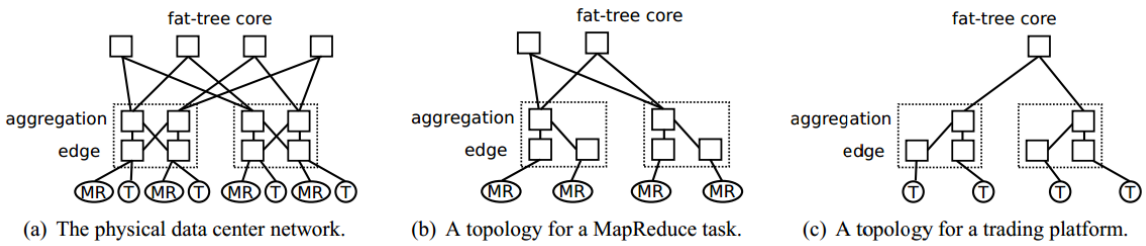

Figura 3. Ottimizzazione delle tipologie per le singole applicazioni, TS trova un sottografo fat-tree a MapReduce (MR) e un spannig tree isolato per la trading platform (T). [9]

TS consente alle applicazioni di creare topologie di rete su misura per soddisfare le loro specifiche esigenze. Consideriamo il fat-tree data center network della Figura 3(a), che collega otto host fisici. Vari sistemi emergenti di routing multi-path supportano le reti well-provisioned, ma i loro sistemi di routing unificati restano ciechi di fronte le esigenze delle singole applicazioni. Tali reti possono ospitare una vasta gamma di applicazioni all'interno di gruppi distinti di VM, come ad esempio la bandwidth-hungry MapReduce/Hadoop cluster (MR) o la trading platform (T) [9]. Mentre le applicazioni MapReduce sono generalmente bottlenecked dalla bandwidth disponibile della rete, le treading platform richiedono prestazioni lowlatency costanti dei percorsi di rete isolati. Topology Switching network tratta queste due applicazioni come due operazioni distinte di routing. Ogni operazione di routing esegue un'istanza di un particolare routing system che meglio affronta il modello di comunicazione e le preferenze di quello specifico compito. Un sistema di routing include un allocator che determina il sottoinsieme della rete fisica che collegherà gli endpoint dell'applicazione nel suo task. In questo caso, il compito di MapReduce [Figura 3(b)] è cercare il percorso di high-bandwidth fisica tra

mappers e reducers per ottimizzare le prestazioni della fase di riordino. Al contrario, la trading platform [Figura 3(c)] assegna per l'isolamento la costruzione di uno spanning tree di proprietà esclusiva. I sistemi di routing definiscono anche un insieme di regole di selezione del percorso, che permettono, per applicazioni specifiche, ai vari switch di prendere decisioni di inoltro su più percorsi.

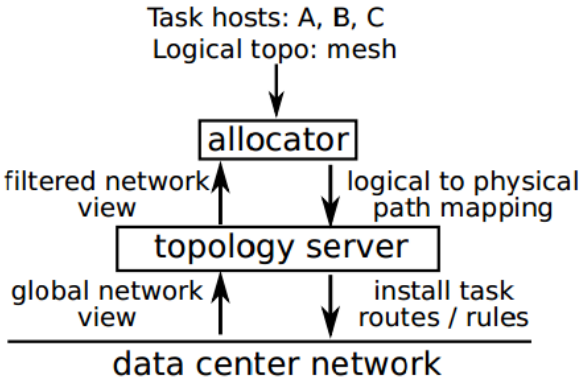

Figura 4. Un topology server media l'accesso alla rete, compilando individual task e performando il controllo d'accesso. [9]

La Figura 4 illustra il processo di compilazione dei compiti di routing in una rete TS. Questi compiti specificano il set di comunicazione del host all'interno del data center, una topologia logica desiderata da costruire tra questi host, e un sistema di routing per gestire l'allocazione dei link e il percorso da selezionare. Gli allocator della topologia prendono come input un set di nodi, un topologia logica desiderata, e una vista delle connettività della rete fisica del topology server. L'allocator quindi associa uno o più percorsi fisici per ogni link della topologia logica per raggiungere gli obiettivi delle performance. L'allocazione avviene on-line, e i singoli allocator non sono autorizzati a cambiare le posizioni esistenti.

Il topology server influenza l'allocazione in molti modi. In primo luogo, dal punto di vista della rete fisica, il server può togliere i collegamenti con link e switch prima di passarlo all'allocator. Questo meccanismo rende banale separare il traffico fisico dalle altre task. Per esempio, il topology server può rimuovere i link di spannig tree [Figura 3(c)] dalla schermata di rete all'allocator di MapReduce. Il server può anche eseguire il controllo di ammissione sulle task, rifiutando d'implementare le task di routing o di revocare le task istanziate. Inoltre questa flessibilità dà agli amministratori un meccanismo tramite il quale aggiornare la rete fisica tra i task allocation.

2.6 Gatekeeper

Rodrigues definisce Gatekeeper come un servizio che cerca di dare l'illusione all'affittuario di un unico switch nonblocking, che collega tutte le sue macchine virtuali [10].

Per ogni VM viene stabilita la garanzia di bandwidth specifica per lo switch sopracitato. Opzionalmente la bandwidth massima può essere impostata più grande della sua garanzia, per consentire l'utilizzo delle larghezze di banda altrimenti sottoutilizzate. Ciò consente al provider di trovare un compromesso tra efficienza e prevedibilità, regolando una o entrambe al minimo o al massimo della bandwidth.

Gatekeeper controlla l'uso di ogni link di accesso al network server. Fornisce per ogni link di scheda di rete virtuale (vNIC) e garanzie di banda in entrambe le direzioni del link network ad ogni server fisico, sia per l'ingresso sia per l'uscita del traffico. Le garanzie minime di bandwidth vengono raggiunte mediante un meccanismo di controllo di ammissione che limita la somma delle

garanzie alla banda del link fisico disponibile. Ogni scheda di rete virtuale può superare la sua allocazione prestabilita, quando la larghezza di banda extra è disponibile sia a trasmettere che a ricevere gli endpoint. Tuttavia, Gatekeeper limita ogni banda vNIC ad un tasso massimo. Configurando un tasso massimo, l'amministratore del sistema può determinare un compromesso per l'efficienza. Per lo scheduling transmission bandwidth, il Gatekeeper usa un tradizionale Weighted Fair Schedule[13], il quale fornisce le garanzie minime di bandwidth. Per il controllo di ricezione di banda, esso monitora il tasso di ricezione del traffico per ciascuna vNIC e il link fisico, e determina l'allocazione della bandwidth di ricezione su ciascun vNIC ad intervalli periodici, prendendo in considerazione l'uso del link, nonché il minimo e il massimo tasso per ogni vNIC. Se la vNIC riceve banda in eccesso alla sua assegnazione, Gatekeeper invia un messaggio di feedback per le altre istanze delle macchine virtuali del hosting.

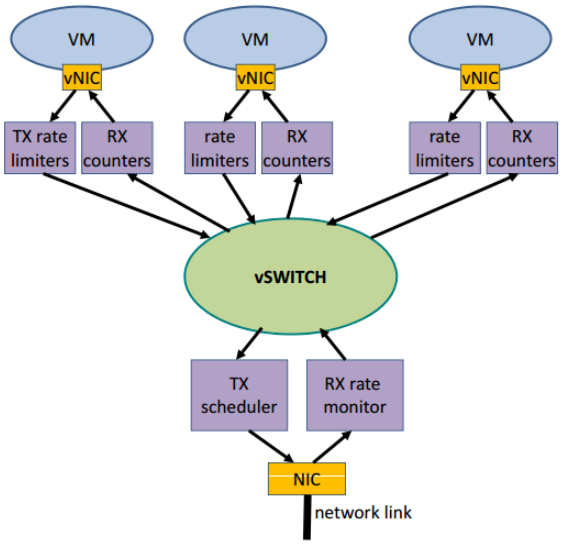

Figura 5. Architettura del Gatekeeper. [10]

[13] E' una tecnica di distribuzione dei flussi che evita la congestione. Controlla direttamente le code dei nodi attraverso un trattamento differenziato del traffico per una determinata disciplina di servizio.

La Figura 5 mostra una panoramica dell'architettura del Gatekeeper. Essa ha un insieme di limitatori di frequenza, associati a ciascuna interfaccia vNIC. Un limitatore di root impone la velocità massima di trasmissione di ogni vNIC. Additional rate limiter è creato dinamicamente per ridurre il tasso del traffico inviato al vNIC di congestione remota. Un pacchetto di filtri classifica i pacchetti in uscita in base al loro indirizzo MAC e lo indirizza all'appropriato rate limiter. In assenza di notifiche di congestione, il rate limit viene incrementato a intervalli periodici secondo una funzione lineare, e se raggiunge il rate massimo del limiter, viene rimosso dopo un intervallo di timeout.

Gatekeeper conserva anche un set dinamico di contatori associato alla vNIC. Questi ultimi misurano il rate tra coppie di vNIC comunicanti. I contatori vengono creati e cancellati in modo dinamico sul set attivo delle vNIC remote inviando il traffico al corrispondente vNIC locale. Ogni contatore memorizza anche l'indirizzo MAC del corrispondente vNIC remoto che viene utilizzato per inviare messaggi di feedback. I contatori vengono creati quando i pacchetti, provenienti da nuove sorgenti, sono ricevuti e cancellati dopo il timeout. Periodicamente, i tassi misurati sono utilizzati per determinare nuovi rate, e in caso di necessità i messaggi feedback di congestione vengono generati.

Generiamo un messaggio di congestione se il rate aggregato sul collegamento fisico supera una determinata soglia o altrimenti se una vNIC supera il suo rate massimo. Se il rate aggregato supera la soglia, viene generato per la vNIC un feedback di congestione che va oltre il rate di ricezione garantito. Gatekeeper fissa il rate della vNIC che vorrebbe ricevere al fin di ottenere una garanzia minima, distribuendo questo rate tra i mittenti attivi. Specifichiamo che un mittente è considerato attivo se il suo rate misurato supera una certa soglia.

Quindi, un messaggio feedback di congestione viene inviato a ciascun mittente attivo con questo rate. Il mittente utilizza queste informazioni per calibrare la

funzione di aumento del rate, tale che esso risulti indipendente dal numero di mittenti.

2.7 Oktopus

Ballani e al. [4] si concentrano sulle performance predictability. Iniziano con un'astrazione "Virtual Cluster", che è simile al Gatekeeper, in cui tutte le macchine virtuali di un affittuario sembrano essere collegate su di un unico switch, con link di capacità B. In seguito lo estendono con il modello "Virtual Oversubscribed Cluster" (VOC), in cui i cluster con bandwidth switch-to-VM di capacità B sono interconnessi con un fattore oversubscribed fissato O. VOC riconosce che le applicazioni hanno una struttura, e rispetta le applicazioni che non necessitano della larghezza di banda massima, tra tutte le coppie di macchine virtuali.

Oktopus è un'implementazione del modello VOC. Esso usa rate limiter hypervisor-based con un algoritmo dettagliato per il collocamento delle macchine virtuali, al fine di soddisfare la richiesta di banda domandata (oppure per rifiutare le richieste che non possono essere soddisfatte).

Il provider gestisce un data center contenente macchine fisiche con slot nel quale le macchine virtuali dell'affittuario possono essere collocate. L'affittuario richiede VM, potendo optare per un cluster (virtuale) o un oversubscribed cluster (virtuale) per connettere le loro VM. Inoltre, per poter consentire la distribuzione incrementale, sosteniamo che gli affittuari che non vogliono una rete virtuale, indicati con lo status quo di chi vuole solo qualche quota delle risorse di rete. Due componenti principali sono utilizzati per ottenere:

- ***Management plane.*** Un network manager (NM) logicamente centralizzato, dopo aver ricevuto una richiesta dell'affittuario, esegue il controllo d'ammissione e mappa le richieste di macchine fisiche.
- ***Data plane*.** Oktopus utilizza rate-limiting agli hypervisor end-host per far rispettare la larghezza di banda disponibile per ogni VM. Questo procedimento assicura che non sia richiesta alcuna prenotazione di banda esplicita agli switch dei data center.

In network manager implementa algoritmi di allocazione per assegnare, on-line, slot su macchine fisiche su richiesta dell'affittuario. Per le richieste che coinvolgono una rete virtuale invece, il NM deve garantire che la richiesta di bandwidth possa essere soddisfatta, ottimizzando il numero di affittuari simultanei. Per ottenerlo, il NM mantiene le seguenti informazioni:

1) La topologia del data center network.
2) La bandwidth residua per ogni link della rete.
3) Gli slot vuoti su ogni macchina fisica.
4) Le informazioni di allocazione per gli inquilini esistenti, incluse le macchine fisiche che vi sono distribuite.
5) I percorsi di rete tra queste macchine e la larghezza di banda riservata ai link lungo questi percorsi.

Il NM assicura che i link che collegano la machine virtuali di un affittuario dispongano di bandwidth sufficiente. Al di là di questo, Oktopus include altri meccanismi per far rispettare le reti virtuali di un affittuario.

2.7.1 Rate limiting delle VM

Le VM individuali non dovrebbero essere in grado di superare la larghezza di banda specificata nella topologia virtuale. Anche se questo potesse essere

realizzato, utilizzando le prenotazioni di banda esplicite a switch, il numero limitato di classi di prenotazione su commodity switch implica che una soluzione del genere di certo non scali con il numero degli affittuari.
Invece, Oktopus si basa sulla velocità di esecuzione basata sugli end-host. Per ogni macchina virtuale su una macchina fisica, un modulo di esecuzione risiede nel OS hypervisor. L'intuizione è che, data la topologia virtuale di un affittuario e il suo traffic rate, è possibile calcolare il rate con cui le coppie di macchine virtuali dovrebbero comunicare. Per raggiungere questo obbiettivo, il modulo di esecuzione per una VM misura il traffic rate delle altre VM. Queste misurazioni del traffico per tutte le VM di un affittuario vengono periodicamente inviate a una singola VM di un affittuario che sarà designato come controller delle VM. Il modulo di esecuzione del controller calcola la giusta quota max-min per il traffico tra le VM. Questi rate saranno trasmessi alle altre VM dell'affittuario, dove il modulo di esecuzione utilizza rate limiter per farle rispettare.
Questo procedimento è simile ad altri meccanismi di controllo dei distributed rate come DRL e Gatekeeper.
La conoscenza della topologia virtuale rende più facile determinare le bottleneck del traffico. Inoltre, il calcolo è tenant-specific, che riduce la portata del problema e ci permette di calcolare i rate per ogni VM indipendentemente.

2.7.2 Affittuari senza reti virtuali

Il traffico della rete per gli affittuari senza risorse garantite, dovrebbe ottenere una quota della bandwidth del link residua nella rete fisica. Ciò si ottiene mediante la priorità a due livelli, e dal commodity switch che offre l'inoltro prioritario. Il traffico proveniente da un affittuario con una rete virtuale è contrassegnato con alta priorità, mentre il resto del traffico con bassa priorità.

Questo, insieme ai meccanismi spiegati in precedenza, assicura agli affittuari con reti virtuali, di ottenere la topologia virtuale e le bandwidth che richiedono, mentre gli altri affittuari ottengono la loro quota di capacità di rete residua.
Con questa implementazione, se una VM appartenente a una rete virtuale non utilizza appieno la sua quota di bandwidth, la capacità rimanente può essere utilizzata solo dagli affittuari senza reti virtuali. Oktopus può utilizzare i meccanismi di condivisione ponderata, per garantire che la capacità inutilizzata sia distribuita tra tutti gli affittuari e quindi che fornisca garanzie minime di banda al posto di garanzie precise.

2.8 FairCloud

Popa e al. [2] nel paper FairCloud, sostengono che le reti cloud dovrebbero fornire, oltre alle garanzie minime di bandwidth, un alta utilizzazione dei link della rete in presenza di richieste non soddisfatte. Esse dovrebbero garantire inoltre la proporzionalità della rete e la divisione della banda tra gli affittuari in proporzione alle loro VM.
Per loro le reti cloud dovrebbero soddisfare i seguenti requisiti:

1) Il primo requisito è fornire agli affittuari garanzie sulla bandwidth minima di rete che possono avere per ogni VM, indipendentemente dall'utilizzazione della rete da parte degli altri affittuari.
2) Il secondo requisito desiderato, si riferisce all'elevato utilizzo dei link. Esso mira a massimizzare l'utilizzo della rete in presenza di richieste insoddisfatte. Ad esempio, vorremmo che un'applicazione usasse l'intera bandwidth disponibile, quando nessun'altra applicazione è attiva. Questo può migliorare in modo significativo le prestazioni per le

applicazioni con modelli di traffico bursty[14], come MapReduce, spiegato nel Topology Switching.

3) L'ultimo requisito è che le risorse di rete sia distribuite tra gli affittuari in proporzione ai loro pagamenti, come avviene per la CPU e la memoria. Il modello di pagamento flat-rate[15] permette che due inquilini con lo stesso numero di VM abbiamo la stessa bandwidth aggregata, ipotizzando che entrambi abbiano richieste sufficienti, avendo pagato la stessa quantità di denaro. Ci riferiamo a questo requisito di assegnazione come la proporzionalità di rete. Popa e al. fanno notare che il requisito minimo di garanzia delle performance non raggiunge la proporzionalità di rete, in quanto si riferisce solo alla garanzia minima della bandwidth delle VM. Tuttavia una VM può ottenere una ripartizione inferiore alla sua garanzia in caso di una diminuzione della domanda, o di una ripartizione maggiore quando le altre VM hanno esigenze minori rispetto alle loro garanzie.

Purtroppo, nessuno dei tradizionali criteri di condivisione della rete -equità tra i flussi, coppie sorgente-destinazione- possono soddisfare una delle garanzie minime o i requisiti di proporzionalità della rete, mentre al contrario proposte come Oktopus possono solo fornire garanzie minime.
La difficoltà di trovare soluzioni per raggiungere tali requisiti deriva dai seguenti compromessi fondamentali:

- C'è un compromesso difficile tra le garanzie minime e la proporzionalità di rete: se si mira a raggiungere le garanzia minime, non si riesce a raggiungere la proporzionalità di rete, e viceversa.

[14] Il traffico bursty si riferisce ad un modello uniforme di trasmissione di dati, a volte ha un'alta velocita di trasmissione dati, altre volte potrebbe essere molto bassa
[15] Tariffa unica

- Anche senza richiedere le garanzie minime, vi è un compromesso tra proporzionalità di rete e l'elevato utilizzo.

2.8.1 Compromesso tra proporzionalità di rete e garanzie minime.

Gli autori mostrano che c'è un compromesso tra il conseguimento della proporzionalità della rete fornendo a ogni VM un'utile garanzia di banda [2].

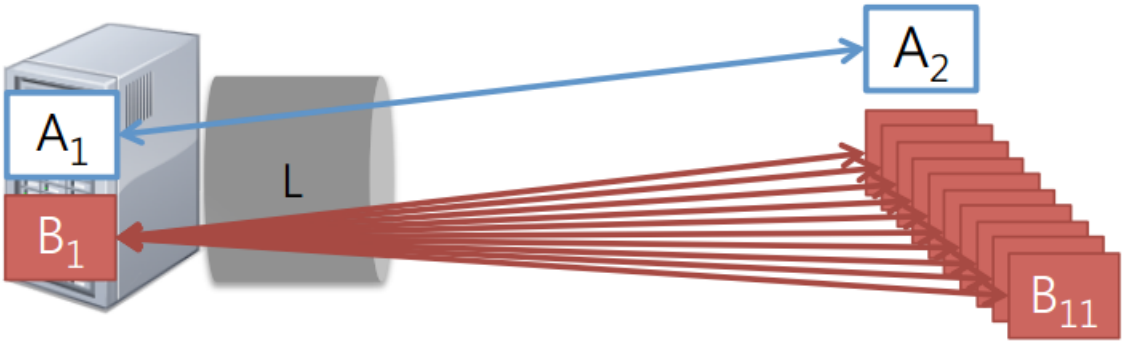

Figura 6. Proporzionalità della rete vs. minime garanzie. Con la proporzionalità, così come B1 comunica con più VM, la bandwidth di A1 può essere diminuita arbitrariamente. [2]

Per illustrare questo compromesso, si considera l'esempio della Figura 6, che mostra due affittuari A e B. A impiega due VM, mentre B impiega undici VM. Le VM A1 e B1 sono ospitate sulla stessa macchina, A1 comunica con A2, mentre B1 comunica con le restanti dieci VM di B. Partiamo dal presupposto che il link di accesso di questa macchina sia l'unico link congestionato del sistema. Secondo il requisito di proporzionalità di rete, A1 dovrebbe ottenere 2/13 dei link di accesso, perché A ha due macchine virtuali e ci sono 13 macchine virtuali totali, mentre B1 dovrebbe ottenere 11/13. Purtroppo, si può arbitrariamente ridurre la bandwidth di A1, semplicemente aumentando il numero di VM che comunicano con B1. In senso stretto, fornisce ancora un

minimo di garanzia, poiché le VM del data center sono finite, la garanzia che ne risulta è troppo bassa per essere utile nelle pratica.

Se consideriamo solo il requisito di garanzia minima, e assumiamo che nessun altra VM può essere ammessa al server, sia A1 che B1 devono aver garantito la metà della capacità del link di accesso. Questa garanzia non deve essere influenzata dal traffico degli altri affittuari nella rete. Tuttavia, come illustrato nell'esempio, vi è un compromesso difficile tra proporzionalità di rete e garanze minime: si può ottenere o la proporzionalità di rete o le garanzie minime, ma non entrambi.

2.8.2 Compromesso tra la proporzionalità di rete e l'elevato utilizzo

Popa e al. mostrano che, anche in assenza del requisito minimo di garanzia, la proporzionalità di rete è difficile da raggiungere [2]. Si dimostra che vi è un compromesso tra la proporzionalità di rete e un elevato utilizzo.

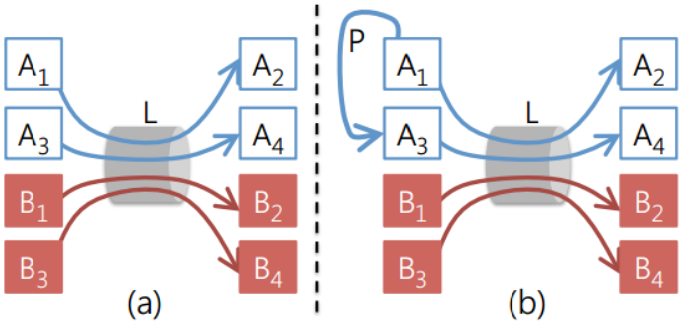

Figura 7. Proporzionalità della rete vs. elevato utilizzo. (a) Le VM degli affittuari A e B hanno quote uguali sul link L. (b) Se A1 inizia a comunicare con A3, l'allocazione di A su L è in diminuzione, quindi A può essere incentivata a non utilizzare il percorso libero P. [2]

Per illustrare questo compromesso, consideriamo l'esempio della Figura 7 che raffigura gli affittuari A e B, dove ciascuno dà lavoro a quattro macchine. La

Figura 7(a) mostra uno scenario in cui i flussi attraversano lo stesso link congestionato L di capacità C; ogni affittuario ottiene C/2 della bandwidth aggregata, e così facendo il requisito della proporzionalità della rete è soddisfatto. Supponiamo ora che le macchine virtuali A1 e A3 iniziano a comunicare lungo un percorso non congestionato P [Figura 7(b)]. Al fine di preservare la proporzionalità di rete, abbiamo bisogno di diminuire l'allocazione dell'affittuario A lungo in link L. Se A ritiene che il suo traffico su L sia più importante che quello tra A1 e A3, A non è incentivata a utilizzare il percorso P, che danneggia l'utilizzo della rete. Faremo riferimento alla capacità di un affittuario di utilizzare un percorso non congestionato senza essere penalizzati su un altro. Quindi, il compromesso tra proporzionalità della rete e l'elevato utilizzo può essere ridotto ad un compromesso tra proporzionalità della rete e gli incentivi all'utilizzazione.

2.9 Location Indipendence

Ballani e al. [12] affrontano il problema di fornire delle adeguate performance di rete in modo alternativo, si concentrano su come garantire che i costi dell'affittuario, nonostante le variabili performance delle rate, non dipendano dalla location delle loro VM. E' importante non solo fornire buone performance, ma bisogna anche considerare come questo influenzi i costi totali a carico dell'affittuario.

Le performance della rete possono dipendere dalla sua location, cioè dove sono allocate le VM dell'affittuario e le altre VM vicine nella rete, qual è il loro carico di lavoro, qual è il protocollo di trasporto usato, etc.

La location delle VM nei data center è un problema per i provider, e non interessa agli affittuari. Ma dato che la location dell'affittuario può influenzare le prestazioni, la situazione diventa ingiusta.

Il modello che loro propongono, avrà per ogni VM una bandwidth aggregata di rete dedicata con le altre VM dello stesso affittuario, così esse potranno trasmettere di più della bandwidth base. In questo modo, l'affittuario pagherà una quota fissa quando genera traffico minore della bandwidth base, mentre, se la supera, i costi saranno proporzionali all'utilizzo. Tale Dominant Resource Pricing (DRP) assicura che i costi non dipendano dalle loro performance di rete, e quindi indipendentemente dalla location. Garantendo un limite inferiore alla bandwidth della rete per ogni istanza, DRP limiterà le performance delle loro applicazioni. Il progetto DRP coinvolge le tecniche esistenti per il posizionamento intelligente delle VM e reti fair queuing[16].

Questo progetto porta a migliorare gli interessi sia del provider che dell'affittuario. I provider sono incentivati a migliorare le performance cosi da massimizzare la capacità di trasmissione del sistema. In questa prospettiva aumenteranno le entrate del provider e probabilmente i costi dell'affittuario.

La realizzazione del DRP deve soddisfare due obiettivi di progettazione principali:

1. Garanzie per la bandwidth base. Il raggiungimento dell'indipendenza della posizione con DRP richiede minori modifiche alla semantica dello IaaS. Ogni VM può inviare e ricevere facendo riferimento alla bandwidth base, e può essere visto come se fosse connesso al resto del data center da un link virtuale con la capacita B_{base}. Si nota che due VM inviano alla stessa VM di destinazione, che potrebbe essere una destinazione bottlenecked; in questo caso è garantito solo un rate combinato di B_{base}

[16] E' un algoritmo di scheduling usato dagli scheduler di rete, per consentire a più flussi di pacchetti di condividere equamente la capacità del link.

invece di $2B_{base}$. Di conseguenza, è possibile che questo modello di traffico potrebbe causare il mancato utilizzo della bandwidth totale.

2. Work conserving allocation. Gli affittuari non dovrebbero essere limitati solo dalla bandwidth base, qualsiasi capacità inutilizzata deve essere usata dai flussi di rete delle VM, che possono inviare al rate più elevato.

Per garantire la bandwidth di base, l'allocazione delle loro VM per le macchine fisiche dovrebbero far corrispondere i loro requisiti di banda. L'idea chiave è di assicurare che ogni link della rete che collega le VM abbia una capacità sufficiente per soddisfare le garanzie della bandwidth. La bandwidth necessaria per un affittuario su un link dipende dal numero di VM che essa ha su qualsiasi lato del link. La quota della bandwidth può essere usata per l'allocazione delle VM, come spiegato in Oktopus [4]. Questo meccanismo è usato dal DRP per allocare VM e soddisfare l'obiettivo 1. Il secondo obiettivo richiede che i flussi, che possono utilizzare una scorta per distribuire la capacità di riserva di un link, diano agli altri affittuari una quota della riserva che essi hanno su quel link. Consideriamo un link di capacità C e un insieme A di affittuari, tramite cui le VM di A inviano traffico. L'allocazione proporzionale implica che i flussi per un affittuario i con quota di bandwidth Qi su questo link, dovrebbe raggiungere un rate aggregato dato da: (Quota della bandwidth + Quota proporzionale della capacità della riserva) = $Qi + (C - \sum_A Qj) * \frac{Qi}{\sum_A Qj} = \frac{CQi}{\sum_A Qj}$.

L'analisi dimostra che la distribuzione della capacità totale del link -in proporzione alle quote dell'affittuario per il link- assicura che le garanzie di bandwidth per le singole VM siano soddisfatte, e che la capacità di riserva sia distribuita in modo paritario. Questa allocazione della bandwidth del link può essere raggiunta con il Weighted Fair Queuing (WFQ). In particolare, gli switch della rete sono configurati con dei pesi per affittuario su ogni sua porta.

Il peso per ogni affittuario e la sua quota sul link in uscita, come determinato dall'algoritmo della allocazione delle VM, si applica a tutto il traffico delle VM dell'affittuario. Questo approccio permette a qualsiasi capacità non utilizzata di essere distribuita tra i flussi che possono utilizzarli. Assumendo una corretta esecuzione del WFQ con queues per affittuario, ciò assicura che, a prescindere dal protocollo di trasporto utilizzato (UDP o TCP), il traffico di un affittuario non può influenzare negativamente il traffico di altri affittuari.

2.10 Choreo

LaCurts e al. [13] intendono agire al application-level, anziché suppore che il provider offra delle garanzie specifiche sulla bandwidth. Analizzano l'applicazione per trovare le sue esigenze di rete, misurano la rete del cloud provider per scoprire la bandwidth disponibile tra le coppie di VM, così l'applicazione posiziona il proprio carico di lavoro per ottimizzare le performance previste. Ipotizzano che, misurando i rate internode e la bottleneck della rete del data center, e analizzando l'applicazione per comprendere le caratteristiche del trasferimento dei dati, è possibile migliorare le performance di diverse applicazioni. Il contesto a cui si riferiscono è di un cliente con una serie di applicazioni cloud network-intensive. L'obiettivo principale è ridurre al minimo la fase di esecuzione dell'applicazione. Per le applicazioni network-intensive, la soluzione ideale è quella di mappare le task dell'applicazione delle VM che prendono le richieste network inter-task, così come il rate network inter-VM preso in considerazione.

Choreo è un sistema di posizionamento della network-aware, i clienti possono utilizzarlo per mettere sul cloud una serie di applicazioni. Choreo ha tre sotto-sistemi:

1. Un componente di misura low-overhead per ottenere i rate del network inter-VM.
2. Un componente per analizzare le caratteristiche del trasferimento dei dati di un'applicazione distribuita.
3. Un algoritmo per mappare le task delle applicazioni delle VM, in modo che le task che comunicano spesso siano inserite su VM con rate più alti tra loro.

Questi sotto-sistemi devono superare i seguenti ostacoli:

1. I rate network inter-VM non sono costanti.
2. I cloud provider spesso utilizzano un modello flessibile per controllare il rate massimo in uscita da qualsiasi VM.
3. Qualsiasi misurazione pratica o metodo di profiling non devono introdurre molto traffico in più.

Un qualsiasi metodo di posizionamento ottimale, dati i rate della rete e il profilo dell'applicazione, è computazionalmente irrisolvibile, quindi qualsiasi approccio pratico può essere solo approssimativo.

Choreo utilizza uno strumento di monitoraggio della rete, come sFlow o TcpDump per raccogliere i modelli di comunicazione delle applicazioni. Esso assume che, da un profiling dell'applicazione offline, verrà a conoscenza -sufficientemente precisa- del modello di comunicazione online dell'applicazione. Choreo comunica le inter-task della rete necessarie di un'applicazione che analizza il numero di byte inviati, piuttosto che il rate osservato. Il motivo di questa particolarità è che il rate della rete ottenuto dipende da cosa sta accadendo nella rete, mentre il numero di byte di solito è indipendente dal traffico incrociato nelle applicazioni cloud.

2.11 Proteus

Xie e al. [14] hanno analizzato diverse applicazioni in stile MapReduce ad alta intensità di dati e hanno dimostrato che molte di queste presentano un comportamento prevedibile, variabile nel tempo, in scale temporali dell'ordine di decine di secondi. Essi propongono l'astrazione Temporally-Interleaved Virtual Cluster (TIVC), simile al VOC, ma che, a differenza di quest'ultimo, consente al provider di ammettere più processi sulla stessa bandwidth.

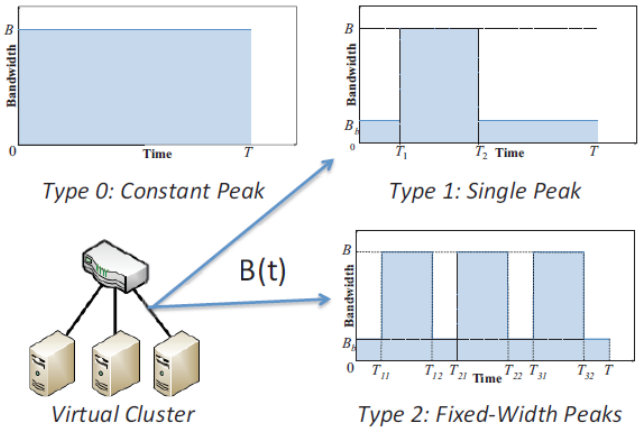

Figura 8. Modello TIVC. [14]

L'astrazione del TIVC consiste in un cluster virtuale di N nodi collegati a uno switch, con link di bandwidth B, di natura simile al VC proposto nel Oktopus. La differenza fondamentale è che la bandwidth per ogni link è una funzione time-varying B(t) invece di un valore costante, come nel lavoro precedente. Questo permette di ottenere il requisito attuale della rete in modo molto più preciso, e consente al provider di ottenere una migliore utilizzazione delle risorse del data center.

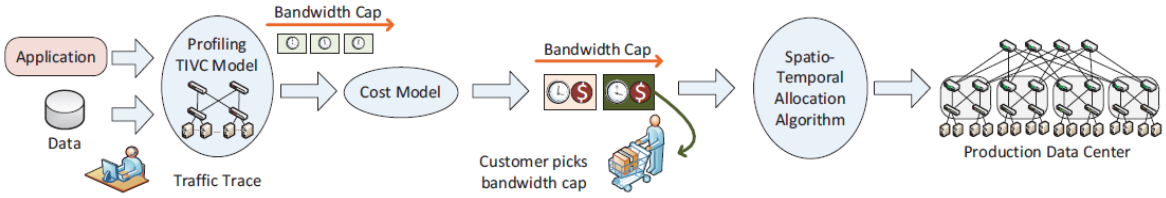

Figura 9. Il sistema Proteus. [14]

Proteus è un sistema che implementa l'astrazione di TIVC. L'obbiettivo di Proteus è quello di consentire ai clienti del cloud di ottenere prestazioni prevedibili e garanzie sul costo per le loro applicazioni. Questo si ottiene in tre fasi, come indicato nella Figura 9.

- Nella prima fase, la domanda del cliente si delinea in diverse configurazioni, cioè secondo le dimensioni dei dati di input per VM e il tetto massimo di bandwidth. Notiamo che il sovraccarico di profiling può essere ridotto drasticamente quando i clienti eseguono ripetutamente gli stessi processi con le stesse dimensioni di input.
- Nella seconda fase, il modello di carica pubblicato dal provider viene utilizzato per stimare il costo per i modelli TIVC candidati sotto diverse configurazioni. Il cliente può quindi scegliere qualsiasi configurazione che meglio si adatta ai suoi obiettivi.
- Nella fase finale, data una configurazione di lavoro TIVC che il cliente sceglie, il provider gestisce un algoritmo spazio-temporale di allocazione del TIVC, per posizionare i processi nel data center fisico, in modo da massimizzare l'utilizzo -e quindi le entrate- del data center cloud, e configura la rete del data center per far rispettare la richiesta di bandwidth variabile nel tempo indicate nelle specifiche del TIVC.

2.12 GRIPhoN

Mahimkar a al. [15] con questa soluzione vogliono proporre un approccio per realizzare dinamicamente una connessione tra data center. GRIPhoN è l'acronimo di Globally Reconfigurable Intelligent Photonic Network e vuole offrire il cosidetto Bod (Bandwidth on demand) ovvero della banda a richiesta all'interno del cuore della rete per una più efficiente comunicazione tra data center. Il concetto alla base del metodo è che esistono diversi tipi di traffico scambiati tra i data center, ovvero quello non interattivo ed effettuato dagli operatori di cloud, e quello generato dagli end user, che invece è di tipo interattivo. Ciò fornisce l'opportunità di diversificare il data rate in base alla funzione da assolvere. Per esempio si potrebbe adoperare un alto data rate (10-40 Gbps) tra data center per i trasferimenti di dati non interattivi, e un basso data rate (1-10 Gbps) per il supporto di sessioni interattive. GRIPhoN fornisce una piattaforma che riesce a realizzare questa connettività dinamica. Il data-rate, che in altre soluzioni risulta statico, qui viene reso dinamico. Si fornisce quindi la possibilità ai provider di regolare la disponibilità di banda tra data center, distanti geograficamente, su richiesta.

Dato che GRIPhoN si basa sul concetto di BoD vediamo i benefici forniti da esso:

1. ***Servizi con data rate configurabile dinamicamente*** selezionando così solo banda in base alle richieste.
2. ***Rapida creazione di nuove connessioni.*** Modifiche di banda dinamici richiedono una connessione rapida. Oggi tutto ciò è realizzabile, a data rate bassi, tramite la riconfigurazione elettronica degli switch [16].
3. ***Ridotto tempo di interruzione*.** In seguito a un qualsiasi errore di rete, è importante ripristinare rapidamente il servizio. Per i servizi con basso

data-rate, i tempi di ripristino sono nell'ordine dei millisecondi. Tuttavia, nessun ripristino è solitamente disponibile oggi per i sistemi "full wavelength".

4. ***Minimo impatto durante la manutenzione***: se si effettuano degli interventi di manutenzione, essi non devono incidere sulle prestazioni. Per tale motivo la gestione della lunghezza d'onda per le connessioni viene effettuata manualmente e non vi è nessun impatto considerevole sul servizio.

GRIPhoN fornisce in sostanza la prima implementazione di BoD che può selezionare il data rate di una connessione con alcune connessioni nell'ordine di 1Gbps e altre che possono svariare dai 10-40 Gbps. Le connessioni "sub-wavelength", ovvero quelle da 1Gbps sono realizzate mediante gli switch OTN (Optical Transport Network), sono presenti nel layer OTN della rete. Le connessioni "Full wavelength", ovvero quelle da 10-40 Gbps, vengono realizzate nel layer fotonico utilizzando dei multiplexer chiamati ROADMs. I provider predispongono degli accessi basati su segnali ottici alla parte core della rete GRIPhoN in più data center e dinamicamente stabiliscono delle connessioni ottiche tra loro.

Risulta di notevole importanza menzionare il componente FXC Fiber Cross-Connect, il quale è designato per instradare i segnali sul layer OTP, o su quello fotonico, ovvero su quello ad alta capacità di trasmissione dati o altrimenti a bassa capacità.

GRIPhoN inoltre offre meccanismi per il wavelength rates di localizzazione dei fault e di ristabilimento delle connessioni. In aggiunta vi è l'automated bridgeand-roll [17] che riduce gli impatti durante la manutenzione.

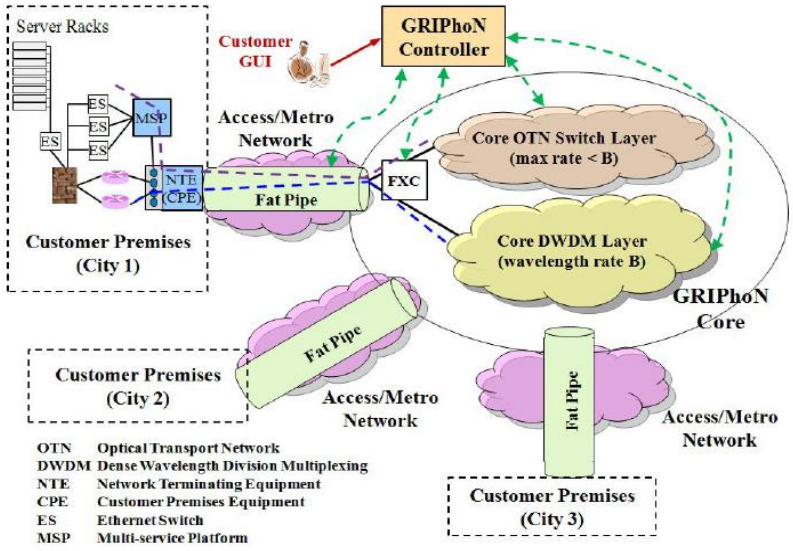

Figura 10. BoD per comunicazioni inter-data center utilizzando GRIPhoN. [15]

Il data center permette connessioni alla rete mediante delle pipe dedicate.

L'instaurazione e il rilascio delle connessioni su richiesta del provider è effettuato dal GRIPhoN controller. Il GRIPhoN controller ha la responsabilità di tenere traccia delle risorse di rete disponibili nel suo database, della comunicazione con gli elementi di rete (controller FXC, OTN switch SME, ROADM SME e controller NTE) al fine di creare o abbattere i collegamenti ordinati dal provider, capacità di gestione delle risorse, gestione del database, rilevamento guasti, della loro localizzazione e del loro ripristino automatico. Come detto in precedenza viene utilizzata la tecnica del bridge-and-roll [17,11] per minimizzare l'interruzione del servizio durante la fase di riconfigurazione della rete o di manutenzione. Con questa tecnica viene creato un nuovo path (the "bridge"), mentre la vecchia connessione viene ancora utilizzata, e verrà dirottato il traffico sul nuovo path quando sarà pronto.

Ogni utilizzatore del servizio possiede una GUI (Graphical User Interface) di GRIPhoN per visualizzare e gestire le connessioni. Inoltre la GUI permette anche di eseguire semplici operazioni di gestione dei fault dal punto di vista

dell'utilizzatore, mostrare lo stato delle connessioni soggette a interruzioni nonché localizzare la posizione dei fault. In sostanza essa maschera tutta la complessità della rete GRIPhoN (access pipes, carrier equipments, network layers, GRIPhoN controller) all'utente.

Capitolo 3

Confronti tra le proposte

Con la seguente disamina si è introdotto il concetto di cloud computing, concentrando l'attenzione sulle performance da esso presentate. Si è appreso che delle carenze di performance si possono ottenere per vari fattori limitanti:

- La disponibilità di banda.
- La memoria fisica.
- La dimensione dei dischi.
- La latenza nelle connessioni.

Ognuno di essi può divenire il bottleneck delle performance o addirittura impedire di usufruire dei servizi. Uno dei fattori più incisivo è rappresentato dalla bandwidth, infatti ove la sua presenza è limitata si riscontra un accesso limitato ai dati e alle applicazioni.

Molti provider cercano di risolvere le problematiche applicative aggiungendo hardware e potenziando l'hardware presente nei data center. Sebbene questa possa essere una soluzione, essa risolve solo in parte il problema delle performance. Considerandone anche la dispendiosità, il rischio in cui si incorre è quello di uno spreco di investimenti economici senza un reale apporto di benefici significativi a livello applicativo soprattutto se comparati allo stesso investimento effettuato.

Assicurare delle performance minime è un aspetto tutt'ora aperto e allo stato attuale esistono una serie di soluzioni adottate che vi riescono più o meno egregiamente.
Nel capitolo precedente si è provveduto a fornire una più ampia visione di alcune di esse, e di seguito si procederà a un loro breve riepilogo e confronto per poter trarre le conclusioni.

DRL: si agisce sulla larghezza di banda della banda globale di un servizio di cloud inserendo dei limitatori di traffico. Dato che i limitatori sono distribuiti in tutta la rete essi hanno bisogno di comunicare tra loro quindi, in questo caso la latenza di comunicazione tra i limitatori, delimita quanto velocemente un limitatore è in grado di adattarsi al cambiamento della domanda in un altro.

SecondNet: in cui si cerca di fornire un livello concordato di servizio. Viene sfruttata la centralità di gestione delle reti di data center e si trova un algoritmo euristico per gestire la suddivisione della banda. Sulla base dei risultati ottenuti, SecondNet risulta realmente una soluzione efficace nel dare una certa garanzia sulla banda. L'assunzione alla base è che ci sia un unico attore centralizzato a gestire la rete dei data center. Il problema dei sistemi centralizzati è che essi rappresentano un cosiddetto "single point of failure" infatti se si riscontrasse un problema nel sistema centralizzato di gestione per esempio della banda si paralizzerebbe, di conseguenza, la piattaforma di cloud.

Seawall: si concentra l'attenzione su un utilizzo efficiente della banda utilizzando dei controller che gestiscono il livello di velocità degli end-host realizzato mediante un sistema di monitoraggio e feedback della rete. L'inserimento dei rate controller concede anche un attenzione all'aspetto della sicurezza in quanto possono per esempio essere limitati gli attacchi di tipo

DDOS. Un aspetto negativo è che non si possono prevedere le prestazioni fornite.

Topology Switching: si fonda sul concetto che diverse applicazioni hanno differenti requisiti di performance. Sono quindi le applicazioni che decidono con che topologia interagire con la rete. L'obiettivo è quello di ottenere una topologia che rispecchi le performance che si desiderano. In termini di scalabilità, Topology Switching può presentare problemi di adattabilità soprattutto se si hanno molti utilizzatori perché risulterebbe difficile trovare un mapping tra i percorsi virtuali e i collegamenti reali di cui si dispone.

Gatekeeper: è una soluzione che cerca di risolvere i problemi relativa alla performance agendo un modo preventivo. Siccome si ha un limite dato dalla capacità di trasmissione del link, si utilizzano dei rate limiter di trasmissione per ogni virtual machine. In questo caso abbiamo che per ogni VM viene garantita una determinata banda. Viene previsto anche un meccanismo di utilizzo di banda che altrimenti risulterebbe inutilizzata. Questa tecnica fornisce un buon compromesso tra efficienza e prevedibilità ma ha alla base un concetto davvero ottimistico ovvero che non avvengano mai congestioni all'interno del core della rete.

Oktopus: è una rivisitazione del Gatekeeper con l'introduzione del modello VOC che cerca di allocare virtual machine per soddisfare le richieste di banda. Proprio questo processo di allocazione di VM può introdurre dei ritardi associati alla riallocazione delle VM al cambio del carico di lavoro da dover essere affrontato.

FairCloud: vengono illustrati una serie di peculiarità che dovrebbe possedere una rete di cloud, arrivando a definire un compromesso, ovvero, che non si possono avere allo stesso tempo garanzie sulla banda e "network proportionality". Viene proposto allora una serie di soluzioni subottimali di cui la maggior parte necessitano di aggiornare l'hardware degli switch ovvero sostenere un costo aggiuntivo.

Le proposte affrontate nel capitolo precedente, e di cui si è appena fornito un breve riepilogo, possono essere tra loro confrontate prendendo in considerazione fattori quali scalabilità, fault-tolerance[17], deployability, le QoS e load-balancing[18].

Scalabilità e fault-tolerance sono questioni importanti per i data center che comprendono un vasto numero di server e di risorse network, e che dovrebbero supportare un gran numero di applicazioni tenant.

Poiché i data center spesso usano commodity server e network hardware, la deployability è una questione chiave che riguarda quanti cambiamenti è necessario apportare all'infrastruttura affinché sia possibile l'implementazione di una particolare architettura.

Allo stesso modo, il QoS è una preoccupazione crescente dei provider in quanto di grande importanza affinché le architetture dei data center virtualizzati siano di successo.

Infine, il load-balancing è un obiettivo importante degli operatori network per la gestione del traffico dati e la riduzione di eventuali congestioni nelle reti.

[17] La tolleranza ai guasti (o fault-tolerance, dall'inglese) è la capacità di un sistema di non subire fallimenti (cioè intuitivamente interruzioni di servizio) anche in presenza di guasti. La tolleranza ai guasti è uno degli aspetti che costituiscono l'affidabilità.

[18] Il load balancing, in italiano bilanciamento del carico, è una tecnica informatica utilizzata nell'ambito dei sistemi informatici che consiste nel distribuire il carico di elaborazione di uno specifico servizio, ad esempio la fornitura di un sito web, tra più server. Si aumentano in questo modo la scalabilità e l'affidabilità dell'architettura nel suo complesso.

Per quanto riguarda la fault-tolerance, la maggior parte delle architetture erano resistenti ai guasti dei componenti data plane[19]. Ad esempio, SecondNet utilizza un canale di segnalazione a forma di albero che permette di rilevare eventuali errori, e il suo algoritmo di allocazione permette poi di gestirli.

L'impatto degli errori nelle architetture con un piano di controllo basato su protocolli spanning tree dipende dal tempo che il protocollo impiega per convergere in seguito alle modifiche della topologia.

Il principale limite di SecondNet è che le sue prestazioni possono dipendere dalla topologia fisica della rete. Inoltre, esso non prende in considerazione altre caratteristiche di prestazione che possono essere importanti per gli utenti, come ad esempio la latenza.

Con SecondNet diviene possibile isolare i VDC (Virtual Data Center), differenziando il servizio e garantendo la bandwidth; la scalabilità può essere realizzata distribuendo la virtualizzazione e la bandwidth nei server. L'algoritmo di allocazione raggiungerà un notevole utilizzo della rete in poco tempo, e l'elasticità sarà capace di contenere l'espansione VDC. Si prospetta che, grazie ad un sistema PSSR (Port-Switching Based Source Routing), SecondNet possa essere applicato a topologie di rete arbitrarie utilizzando commodity services e switches.

A differenza di implementazioni basate sulle prenotazioni statiche, come accade per esempio con SecondNet, l'attuazione di Seawall è una soluzione che permette di raggiungere un utilizzo maggiore sfruttando il multiplexing, con un collegamento condiviso che viene adattato alle esigenze di traffico istantanea dei flussi di dati che vengono trasferiti sui vari canali.

Dal canto suo, Seawall supporta topologie generali, consentendo di fornire benefici anche nelle reti di data center con costi limitati. Tali topologie hanno

[19] Parte della rete che trasporta il traffico degli utenti.

solitamente una bandwidth molto contenuta, tutti i nodi che utilizzano un determinato collegamento principale devono essere stimati per consentire una equa ripartizione, una prenotazione della bandwidth e un controllo della congestione. Nello specifico, Seawall utilizza l'informazione topologica per prevenire il sovrautilizzo del link.

Si comprende come esso consista in un sistema di isolamento delle prestazioni nell'ambito dei network cloud data center che fornisce una notevole capacità di rete tra gli utenti; le principali preoccupazioni sulla distribuzione e il mantenimento delle applicazioni cloud riguardano la variazione delle prestazioni e la disponibilità del servizio, parametri che possono essere migliorati con l'inserimento di supporti speciali nella rete.

La proposta Distributed Rate Limiting, differentemente delle altre, mostra una notevole resistenza al fallimento, e la capacità complessiva di DRL è quella di apportare delle modifiche senza incidere in alcun modo sul QoS. Tale resistenza al fallimento rende maggiormente tollerante anche il cloud fault e, siccome questi sono problemi di natura dinamica, DRL deve essere capace di reagire in modo dinamico con CBSP[20].

Per quanto riguarda il Topology Switching, esso rifiuta l'approccio one-size-fits-all per la selezione del percorso che gli permette di fornire le applicazioni e di entrare quindi a far parte delle decisioni routing. In tal senso, vi sono dei ripartitori che permettono alle applicazioni di sfruttare i percorsi che altri preferiscono non utilizzare, soddisfacendo gli obiettivi di applicazione con una struttura ad albero.

L'utilizzo di più rate limiters consente a un allocative system di non incorrere in successivi problemi, in quanto una loro mancanza comporta delle conseguenze poco piacevoli. Per esempio, un comportamento scorretto

[20] Capacity Building for Service Provider

potrebbe portare Gatekeeper a penalizzare VM non coinvolti solo perché inviano dati alla stessa destinazione. L'utilizzo dunque di rate limiters più complessi migliorerebbe la performance di sistema.

Per massimizzare la performance, un sistema dovrebbe supportare un ampio numero di limiters di varia capacità. L'attuale architettura di Seawall può supportare limiters rate-based o window-based a livello di hardware o di software.

Per quanto riguarda invece la deployability, architetture come Seawall richiedono dei cambiamenti soltanto nel hypervisor, e la maggior parte delle architetture considerate richiedono delle funzioni hardware. In tal senso, bisogna dire che con l'evoluzione dell'hardware e l'adozione di hardware programmabili, tali architetture non vengono escluse da queste tecnologie, ma diventeranno in futuro un luogo comune.

Per ciò che concerne le QoS, SecondNet provvede a garantire l'allocazione della bandwidth per ogni network virtuale. Dal canto suo, Seawall fornisce una equa condivisione della bandwidth tra gli utenti, la differenza, però, è che non garantisce l'allocazione della bandwidth in quanto non vi è alcuna prestazione prevedibile.

Oktopus, ancora, fa parte di quel tipo di architetture che riescono a supportare QoS soltanto in combinazione alle soluzioni che garantiscono la bandwidth.

Insieme a SecondNet, Oktopus ha proposto dei modelli statici in tutta la rete per implementare le garanzie di bandwidth dei modelli flessibili; l'inconveniente principale di tali sistemi è stato il mancato raggiungimento della proprietà di conservazione del lavoro, dato che la bandwidth non utilizzata non viene condivisa tra gli utenti. Del resto, il vantaggio dei sistemi di conservazione è che possono raggiungere delle topologie virtuali maggiormente complesse indipendentemente dalla posizione fisica dei VM.

NetShare, invece, sostiene il network sharing valutando il "peso" di un tenant, e mantenendolo costante per tutta la rete; l'utilità di tale modello si riscontra quando si intende implementare una forma di collegamento proporzionale.

Seawall e NetShare offrono come soluzione la condivisione della bandwidth tra i tenants a seconda del loro peso; la distribuzione proporzionale di bandwidth che ne risulta comporta una maggiore efficacia dell'infrastruttura sottostante; rispetto a Oktopus, però, la performance del singolo tenant continua a essere dipendente da quella degli altri tenant, proprio in virtù della condivisione della bandwidth. Con Oktopus si rendono possibili topologie di maggiore flessibilità che riescono a creare un equilibrio nel trade-off tra le richieste dei tenant e la flessibilità del provider.

Per quanto riguarda Gatekeeper, esso utilizza un meccanismo di supervisione indicato soltanto per i network bisection-bandwidth. Tale soluzione permette di supportare il modello di prestazione nei data center virtualizzati; Gatekeeper ha una valenza notevole in scenari semplici, ma appare ancora poco chiaro il suo comportamento in rapporto a carichi di lavoro di maggior entità e dinamicità.

Per quanto riguarda la location independent, nel caso di lavori con basse prestazioni, diviene una soluzione utile visti i costi inferiori previsti per ogni utente; inoltre, vengono anche allineati gli interessi dell'utente a quelli del provider, incoraggiando quest'ultimo all'innovazione.

Choreo appare una soluzione utile per migliorare le prestazioni a livello di applicazione; grazie alla sua notevole capacità di trasferimento dati, Choreo evita il rallentamento dei processi nella rete, riportando un miglioramento massimo del 61% quando le applicazioni vengono posizionate tutte insieme e del 79% quando esse arrivano in tempo reale.

Grazie a Choreo è stato possibile sviluppare e testare le tecniche per la misurazione delle reti cloud pubbliche in modo rapido e preciso; di

conseguenza è stato possibile posizionare le applicazioni rapidamente, con una precisione tale da rilevare un netto miglioramento rispetto agli altri metodi di posizionamento.

Sebbene Choreo possa migliorare il tempo di completamento di molte applicazioni cloud, non risulta essere una scelta appropriata per ogni applicazione, né per ogni ambiente cloud. In particolare, mostra i suoi limiti soprattutto nel caso di applicazioni che richiedono pochi minuti di caricamento, in quanto i suoi costi di misurazione renderebbero vana qualunque miglioria.

Anche nel caso di applicazioni con uso relativamente uniforme della bandwidth, Choreo non apporterebbe grandi miglioramenti; poiché infatti ogni coppia di VM usa quasi la stessa quantità di bandwidth, Choreo non è in grado di agevolare nell'uso la coppia "più grande". Infine, anche le applicazioni interattive non riporterebbero grandi migliorie, in quanto la versione attuale di Choreo non rappresenta i cambiamenti delle applicazioni nel tempo. Sulla base di quanto detto, pertanto, Choreo si rivela realmente utile solo per determinati ambienti cloud, quelli cioè che non prevedono una rete con bandwidth in abbondanza.

I servizi dinamici sono quelli che pongono maggiori difficoltà alla garanzia di adeguate risorse di rete per il sostentamento della domanda prevista dai provider; per poter fronteggiare una situazione del genere, bisognerebbe pianificare anticipatamente dove e quando utilizzare le risorse. In tal senso, GRIPhoN permette di gestire le diverse risorse, e l'iniziale pianificazione può sembrare simile alla progettazione che viene eseguita in POTS (Plain Old Telephony Services) con risorse statiche che vengono gestite da più utenti. Pertanto, in questo tipo di rete il numero di utenti è minore mentre è maggiore il costo di una linea, motivo per cui diviene critico fornire una progettazione accurata.

Un’applicazione interessante del GRIPhoN risulta particolarmente interessante in virtù della sua tolleranza dei tempi di connessione nell’ambito network grooming; i path disponibili vengono resi maggiormente appropriati grazie ai collegamenti che sono stati precedentemente stabiliti. GRIPhoN dunque può rivelarsi di grande utilità per lo spostamento delle connessioni wavelength.

Capitolo 4

Prospettive future e conclusioni

Le network performance consistono in misure di qualità di un prodotto di telecomunicazione e possono essere sia a commutazione di circuito che di pacchetto, ma in entrambi i casi la prestazione viene misurata rilevando la QoS, insieme ad altri fattori come la trasmissione dati, la stabilità, la tecnica di modulazione, ecc.

Oggigiorno, i servizi cloud-based integrano le risorse distribuite a livello globale nelle piattaforme di calcolo, senza alcuna soluzione di continuità; inoltre, gli attuali modelli di previsioning e accounting non hanno la flessibilità necessaria per supportare in modo efficace la composizione dinamica e il rapido spostamento dei dati nel software.

Nel presente lavoro, partendo dalla definizione, sulla base anche delle informazioni fornite dal NIST, di alcuni punti chiavi del concetto di cloud computing, relativi, in particolare, alle caratteristiche generali, e ai modelli di distribuzione e di servizio, si è insistito molto sulle problematiche relative alle performance degli ambenti cloud, e alle diverse proposte attualmente presenti sul mercato con i relativi limiti.

Dopo averle illustrate in modo dettagliato, sulla base delle informazioni reperibili nella letteratura, le diverse proposte sono state tra loro messe a confronto al fine di evidenziare, per ciascuna di essa, tanto gli aspetti positivi quanto i punti di criticità.

Dal confronto si è avuto modo di mettere in luce come nessuna delle soluzioni proposte possa essere effettivamente considerata valida per tutte le questioni che bisognerebbe prendere in considerazione nel contesto della network dei data center virtualizzati.

Si è rilevato, infatti, che il problema principale per cui tali proposte non possono essere universalmente valide riguarda l'attenzione specifica che esse prestano a un solo aspetto della virtualizzazione dei data center, tralasciandone altri di altrettanta importanza.

Per esempio, se sappiamo essere in presenza di una rete con bassa probabilità di congestione la soluzione Gatekeeper potrebbe essere quella più adatta mentre se abbiamo dei requisiti di forte scalabilità dobbiamo escludere a priori la soluzione Topology Switching. Invece se non abbiamo interesse a prevedere le prestazioni fornite prestando al contempo attenzione alla sicurezza, la soluzione Seawall risulta adatta; mentre se prevediamo un sistema di gestione centralizzato ci si può orientare verso la soluzione SecondNet. Infine il metodo DRL non deve essere utilizzato in casi in cui si richieda un'elevata responsiveness del sistema.

La conclusione a cui si è giunti, quindi, è che allo stato attuale nessuna di queste proposte risulta essere valida in generale, ma piuttosto ognuna singolarmente può rivelarsi valida a seconda dei requisiti specifici che si cerca di soddisfare. Diversamente, dunque, una soluzione auspicabile in futuro potrebbe essere quella di strumenti che, partendo da quelle che sono le potenzialità delle proposte in questa sede analizzate, siano in grado di realizzare una combinazione organica delle loro diverse caratteristiche chiave in modo da poterne sfruttare i rispettivi vantaggi.

In tal senso, la ricerca della migliore combinazione richiederebbe una attenta comprensione dei requisiti di prestazione delle applicazioni che risiedono nel data center.

Bibliografia

[1] Peter Mell, Timothy Grance. The NIST Definition of Cloud Computing (Draft). National Institute of Standards and Technology Special Publication 800-145 (Draft). 2011.

[2] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, and I. Stoica. FairCloud: Sharing the Network in Cloud Computing. Helsinki, Finland. In Proc. SIGCOMM, pagine 187-198, 2012.

[3] Jeffrey C. Mogul, Lucian Popa. What We Talk About When We Talk About Cloud Network Performance. ACM SIGCOMM Computer Communication Review. Volume 42, Number 5, October 2012.

[4] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron. Towards predictable datacenter networks. Toronto, Ontario, Canada. In Proc. SIGCOMM, pages 242–253, 2011.

[5] Barath Raghavan, Kashi Vishwanath, Sriram Ramabhadran, Kenneth Yocum, and Alex C. Snoeren. Cloud Control with Distributed Rate Limiting. Kyoto, Japan. SIGCOMM 2007 Conference.

[6] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang. SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees. Philadelphia, USA. ACM CoNEXT Conference, 2010.

[7] A. Shieh, S. Kandula, A. Greenberg, and C. Kim. Seawall: performance isolation for cloud datacenter networks. Boston, MA, USA. HotCloud Conference, 2010.

[8] C. Hopps. RFC 2992: Analysis of an Equal-Cost Multi-Path Algorithm, 2000.

[9] K. C. Webb, A. C. Snoeren, and K. Yocum. Topology switching for data center networks. Boston, MA, USA. Hot-ICE Conference, 2011.

[10] H. Rodrigues, J. R. Santos, Y. Turner, P. Soares, and D. Guedes. Gatekeeper: supporting bandwidth guarantees for multi-tenant datacenter networks. Portland, OR, USA. WIOV Conference, 2011.

[11] A. L. Chiu, G. Choudhury, G. Clapp, R. Doverspike, J. W. Gannett, J. G. Klincewicz, G. Li, R. A. Skoog, J. Strand, A. von Lehmen, and D. Xu. Network design and architectures for highly dynamic next-generation ip-over-optical long distance networks. In Journal of Lightwave Technology, 2009.

[12] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron. The price is right: towards location-independent costs in datacenters. Cambridge, MA, USA. HotNets-X Conference, 2011.

[13] Katrina LaCurts, Shuo Deng, Ameesh Goyal, Hari Balakrishnan. Choreo: Network-Aware Task Placement for Cloud Applications. Barcelona, Spain. IMC Conference, 2013.

[14] D. Xie, N. Ding, and Y. C. Hu. The Only Constant is Change: Incorporating Time-Varying Network Reservations in Data Centers. Helsinki, Finland. SIGCOMM Conference, 2012.

[15] Ajay Mahimkar, Angela Chiu, Robert Doverspike, Mark D. Feuer, Peter Magill, Emmanuil Mavrogiorgis, Jorge Pastor, Sheryl L. Woodward, Jennifer Yates. Bandwidth on Demand for Inter-Data Center Communication. Cambridge, MA, USA. HotNets-X Conference, 2011.

[16] R. Doverspike. Practical aspects of bandwidth-on-demand in optical networks. In Panel on Emerging Networks, Service Provider Summit. Anaheim, CA, USA. OFC Conference, 2007.

[17] X. J. Zhang, M. Birk, A. Chiu, R. Doverspike, M. D. Feuer, P. Magill,

E. Mavrogiorgis, J. Pastor, S. L. Woodward, and J. Yates. Bridge-and-roll demonstration in griphon (globally reconfigurable intelligent photonic network). San Diego, CA, USA. OFC Conference, 2010.