

IL MODELLO DI PROGRAMMAZIONE WINDOWS AZURE

SOMMARIO

Perché creare un nuovo modello di programmazione?	3
Le tre regole del modello di programmazione Windows Azure	4
Un'applicazione Windows Azure è costituita da uno o più ruoli.	4
Un'applicazione Windows Azure esegue più istanze di ciascun ruolo.	5
Un'applicazione Windows Azure presenta un comportamento corretto in caso di problemi di un'istanza di un ruolo.	6
Che cosa offre il modello di programmazione Windows Azure	8
Concetti di base: il controller di infrastruttura	8
I vantaggi: amministrazione, disponibilità e scalabilità migliorate.....	9
Implicazioni del modello di programmazione Windows Azure: altre modifiche	11
Interazioni con il sistema operativo	11
Interazioni con l'archiviazione permanente	12
Interazioni tra le istanze di ruolo	14
Trasferimento di applicazioni Windows Server in Windows Azure	15
Conclusioni	17
Per ulteriori informazioni	17
Informazioni sull'autore	17

PERCHÉ CREARE UN NUOVO MODELLO DI PROGRAMMAZIONE?

Milioni di sviluppatori in tutto il mondo hanno già familiarità con la creazione di applicazioni tramite il modello di programmazione Windows Server. Tuttavia, le applicazioni scritte per Windows Azure, la piattaforma cloud di Microsoft, non utilizzano esattamente questo modello già noto. Anche se la maggior parte delle competenze di uno sviluppatore Windows continua a essere applicabile, Windows Azure prevede uno specifico modello di programmazione.

Perché? Perché non limitarsi a replicare esattamente nella cloud il già familiare ambiente Windows Server? È quel che fanno le piattaforme cloud di molti fornitori, offrendo macchine virtuali (VM) che operano in locale. Questo approccio, comunemente definito *Infrastructure as a Service (IaaS)*, ha certamente una sua validità e, per alcune applicazioni, rappresenta la scelta ideale. Le piattaforme cloud, tuttavia, costituiscono un ambiente del tutto nuovo, dotato del potenziale per risolvere i problemi di oggi in modi innovativi. Anziché IaaS, Windows Azure offre un'astrazione di livello superiore, generalmente classificata come *Platform as a Service (PaaS)*. Pur essendo per vari aspetti simile all'ambiente Windows locale, questo tipo di astrazione dispone di un proprio modello di programmazione, pensato per aiutare gli sviluppatori a creare applicazioni più efficienti.

Il modello di programmazione Windows Azure si concentra sul miglioramento delle applicazioni in tre aree:

- *Amministrazione. Nelle tecnologie PaaS, è la piattaforma a gestire direttamente la maggior parte delle attività amministrative. In Windows Azure, questo significa che è la piattaforma a occuparsi automaticamente di attività come l'applicazione delle patch di Windows e l'installazione di nuove versioni del software di sistema. L'obiettivo è ridurre le attività e i costi connessi alla gestione dell'ambiente applicativo.*
- *Disponibilità. Con le applicazioni attuali, i tempi di inattività (pianificati o meno) dovuti all'applicazione delle patch di Windows, agli aggiornamenti delle applicazioni, a problemi hardware e ad altre motivazioni, sono un'evenienza del tutto normale. Tuttavia, vista la ridondanza resa possibile dalle piattaforme cloud, non vi è più alcun motivo per accettare questa situazione. Il modello di programmazione Windows Azure è progettato per garantire una costante disponibilità delle applicazioni, anche in caso di aggiornamenti del software e di problemi hardware.*
- *Scalabilità. I tipi di applicazioni che gli sviluppatori desiderano creare per la cloud sono spesso pensati per gestire un numero elevato di utenti. Il tradizionale modello di programmazione Windows Server, tuttavia, non è stato pensato espressamente per supportare le applicazioni basate su Internet. Il modello di programmazione Windows Azure, invece, è progettato a tale scopo fin dall'inizio. Creato per l'era della cloud, questo modello consente agli sviluppatori di realizzare applicazioni scalabili, che possano essere supportate da potenti data center cloud. Altrettanto importante, il modello assicura, quando necessario, la scalabilità verso il basso delle applicazioni, che in questo modo possono utilizzare solo le risorse effettivamente necessarie.*

Che lo sviluppatore utilizzi una tecnologia IaaS o una soluzione PaaS come Windows Azure, la creazione di applicazioni su piattaforme cloud presenta alcuni vantaggi intrinseci. Entrambi gli approcci, ad esempio, consentono di pagare solo per le risorse di elaborazione effettivamente utilizzate ed evitano i tempi di attesa associati alla distribuzione dei server da parte del reparto IT aziendale. Per quanto importanti, questi vantaggi non sono l'argomento principale della nostra discussione. Il nostro obiettivo è invece chiarire che cos'è il modello di programmazione Windows Azure e quali vantaggi offre.

LE TRE REGOLE DEL MODELLO DI PROGRAMMAZIONE WINDOWS AZURE

Per mantenere le sue promesse, il modello di programmazione Windows Azure richiede che le applicazioni rispettino tre regole:

- *Un'applicazione Windows Azure è costituita da uno o più ruoli.*
- *Un'applicazione Windows Azure esegue più istanze di ciascun ruolo.*
- *Un'applicazione Windows Azure presenta un comportamento corretto in caso di problemi di un'istanza di un ruolo.*

Vale la pena sottolineare il fatto che in Windows Azure è possibile eseguire anche le applicazioni non conformi a tutte queste regole. Di fatto, la piattaforma non impone il rispetto di queste regole, ma semplicemente lo presuppone da parte di qualsiasi applicazione. È tuttavia necessario tenere presente che, se si sceglie di eseguire in Windows Azure un'applicazione che viola una o più delle regole, questa non utilizzerà effettivamente il modello di programmazione Windows Azure. Se non si comprendono e rispettano le regole del modello, l'applicazione potrebbe non funzionare come previsto.

UN'APPLICAZIONE WINDOWS AZURE È COSTITUITA DA UNO O PIÙ RUOLI.

Che un'applicazione venga eseguita nella cloud o nel data center aziendale, quasi certamente sarà possibile suddividerla in parti logiche. Windows Azure formalizza queste divisioni in *ruoli*. Un ruolo comprende uno specifico set di codice, ad esempio un assembly .NET, e definisce l'ambiente in cui tale codice viene eseguito. Windows Azure consente agli sviluppatori di creare tre diversi tipi di ruoli:

- *Ruolo Web. Come suggerisce il nome, i ruoli Web sono destinati principalmente alla logica che interagisce con il mondo esterno, attraverso il protocollo HTTP. Il codice scritto come ruolo Web in genere riceve il proprio input attraverso Internet Information Services (IIS) e può essere creato utilizzando varie tecnologie, come ASP.NET, Windows Communication Foundation (WCF), PHP e Java.*
- *Ruolo di lavoro. La logica scritta come ruolo di lavoro è in grado di interagire con il mondo esterno in vari modi, non solo tramite HTTP. Ad esempio, un ruolo di lavoro può includere codice che converte video in un formato standard, calcola il rischio di un portafoglio d'investimento o esegue un determinato tipo di analisi dei dati.*
- *Ruolo VM (macchina virtuale): un ruolo VM esegue un'immagine, ovvero un disco rigido virtuale (VHD), di una macchina virtuale Windows Server 2008 R2. Questo VHD viene creato utilizzando un computer Windows Server locale e quindi caricato in Windows Azure. Una volta memorizzato nella cloud, il disco rigido virtuale può essere caricato ed eseguito su richiesta in un ruolo VM.*

Tutti e tre i ruoli svolgono una funzione specifica. Il ruolo VM, tuttavia, è stato reso disponibile solo di recente, pertanto è corretto affermare che attualmente le opzioni più utilizzate sono i ruoli Web e di lavoro. Nella figura 1 è illustrata una semplice applicazione Windows Azure creata con un ruolo Web e un ruolo di lavoro.



Figura 1: un'applicazione Windows Azure è costituita da uno o più ruoli, ad esempio dalla combinazione di un ruolo Web e di un ruolo di lavoro come in questo caso.

Questa applicazione potrebbe utilizzare un ruolo Web per accettare richieste HTTP da parte degli utenti, quindi delegare a un ruolo di lavoro le attività associate a tali richieste, ad esempio la conversione del formato di un file video in modo da renderlo disponibile per la visualizzazione. Una delle ragioni principali per questa divisione dei ruoli è il fatto che suddividere le attività in questo modo può facilitare la scalabilità di un'applicazione.

Un'applicazione Windows Azure può anche essere costituita da un unico ruolo Web o di lavoro: non è necessario utilizzarli entrambi. Una singola applicazione può anche includere tipi diversi di ruoli Web e di lavoro. Ad esempio, potrebbe includere un ruolo Web che implementa un'interfaccia browser, magari creata utilizzando ASP.NET, e un altro ruolo Web che espone un'interfaccia di servizi Web implementati tramite WCF. Analogamente, un'applicazione Windows Azure che esegue due diversi tipi di analisi dei dati potrebbe definire un ruolo di lavoro distinto per ognuna. Per mantenere il massimo livello di semplicità possibile, tuttavia, partiremo dal presupposto che l'applicazione di esempio qui descritta includa un solo ruolo Web e un solo ruolo di lavoro.

Come parte della creazione di un'applicazione Windows Azure, lo sviluppatore crea un file di definizione del servizio che denomina e descrive i ruoli dell'applicazione. Questo file può anche specificare altre informazioni, ad esempio le porte su cui ogni ruolo rimarrà in ascolto. Windows Azure utilizza queste informazioni per creare l'ambiente corretto per l'esecuzione dell'applicazione.

UN'APPLICAZIONE WINDOWS AZURE ESEGUE PIÙ ISTANZE DI CIASCUN RUOLO.

Ogni applicazione Windows Azure è costituita da uno o più ruoli. In fase di esecuzione, un'applicazione conforme al modello di programmazione Windows Azure deve eseguire almeno due copie, ovvero due istanze distinte, di ogni ruolo in essa contenuto. Ogni istanza viene eseguita in una macchina virtuale distinta, come mostrato nella figura 2.

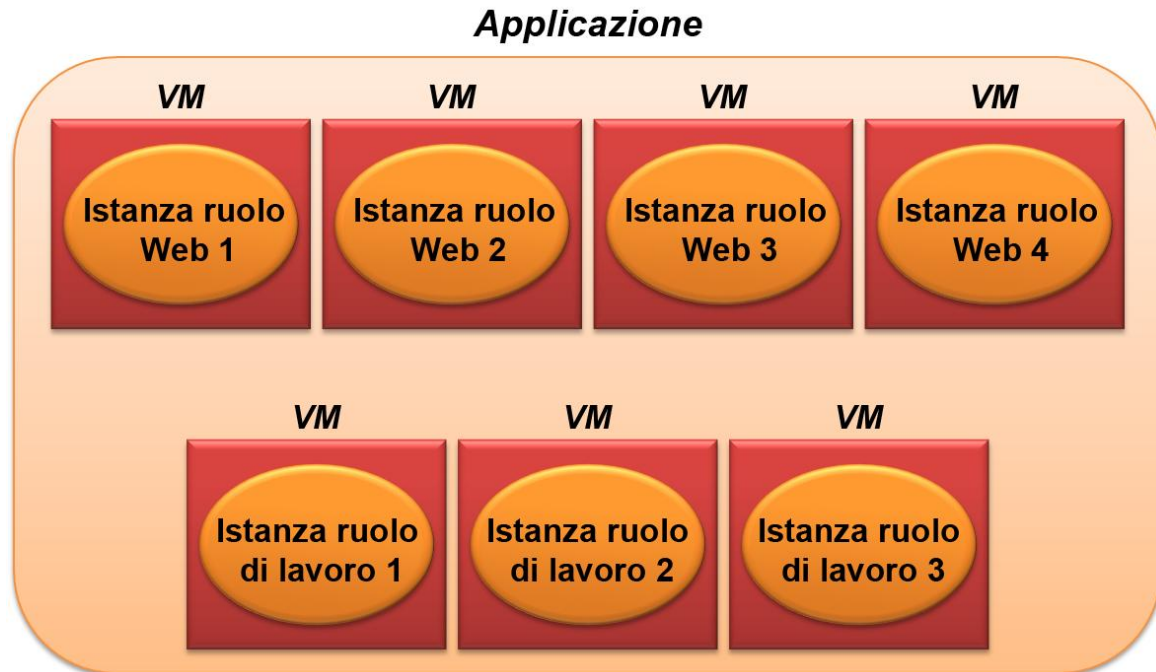


Figura 2: un'applicazione Windows Azure esegue più istanze di ogni ruolo.

Come accennato, l'applicazione di esempio qui descritta include un solo ruolo Web e un solo ruolo di lavoro. Lo sviluppatore può indicare a Windows Azure quante istanze di ogni ruolo eseguire, attraverso un file di configurazione del servizio (diverso dal file di definizione del servizio citato nella sezione precedente). In questo caso, lo sviluppatore ha richiesto quattro istanze del ruolo Web dell'applicazione e tre istanze del ruolo di lavoro.

Ogni istanza di un particolare ruolo esegue esattamente lo stesso codice. Di fatto, con la maggior parte delle applicazioni Windows Azure, ogni istanza è identica a tutte le altre dello stesso ruolo: le istanze sono intercambiabili. Ad esempio, Windows Azure esegue automaticamente il bilanciamento del carico per le richieste HTTP tra le istanze del ruolo Web di un'applicazione. Questo bilanciamento del carico non supporta sessioni permanenti, di conseguenza non è possibile indirizzare tutte le richieste di un client alla stessa istanza di ruolo Web. L'archiviazione dello stato specifico del client, ad esempio un carrello per gli acquisti online, in una particolare istanza di un ruolo Web non funzionerà, dato che in Windows Azure non è possibile assicurare che tutte le richieste di un client siano gestite da tale istanza. Questo tipo di stato deve invece essere memorizzato esternamente, come descritto più avanti.

UN'APPLICAZIONE WINDOWS AZURE PRESENTA UN COMPORTAMENTO CORRETTO IN CASO DI PROBLEMI DI UN'ISTANZA DI UN RUOLO.

Un'applicazione conforme al modello di programmazione Windows Azure deve essere creata utilizzando i ruoli e deve eseguire due o più istanze di ciascuno di essi. Inoltre, deve presentare un comportamento corretto in caso di problemi di una di queste istanze. Nella figura 3 è illustrato questo concetto.

Applicazione

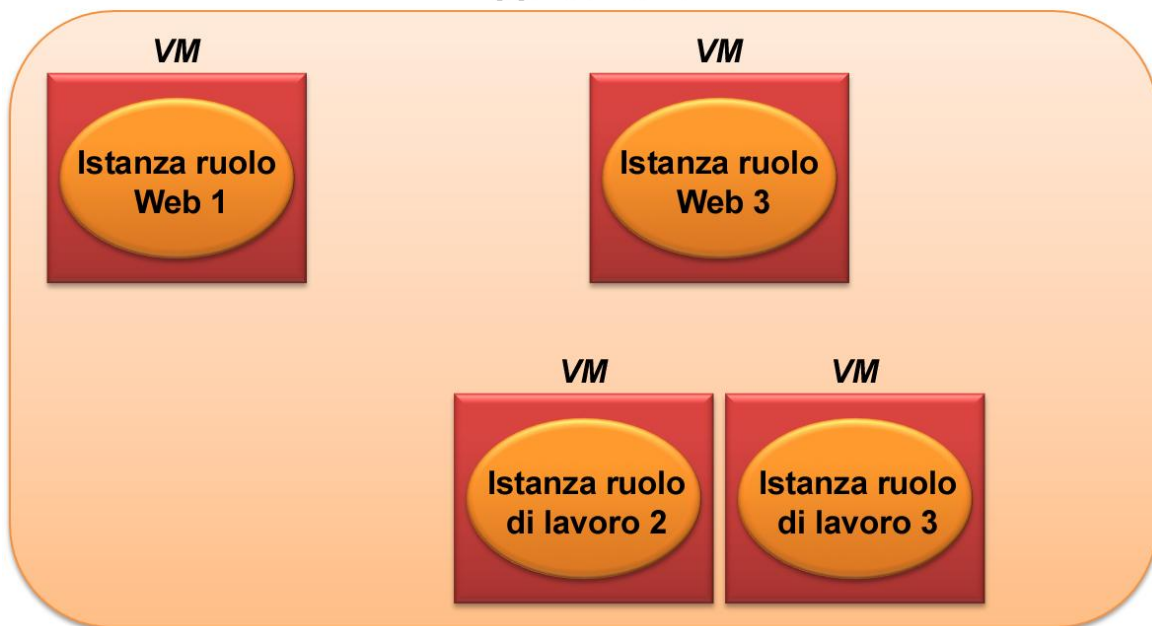


Figura 3: un'applicazione Windows Azure presenta un comportamento corretto in caso di problemi di un'istanza di un ruolo.

In questo caso, l'applicazione illustrata nella figura 2 ha perso due delle istanze del ruolo Web e una delle istanze del ruolo di lavoro. Questo può essere dovuto a problemi dei computer in cui le istanze erano in esecuzione o all'interruzione della connessione di rete fisica. Indipendentemente dall'origine del problema, è probabile che le prestazioni dell'applicazione ne risentano, dato il minor numero di istanze disponibili per svolgere l'attività richiesta. Tuttavia, l'applicazione rimane operativa e continua a funzionare correttamente.

Se tutte le istanze di un particolare ruolo presentano problemi, inevitabilmente l'applicazione smetterà di funzionare correttamente. Tuttavia, il requisito di continuare a funzionare correttamente in caso di interruzione parziale è essenziale per il modello di programmazione Windows Azure. Il contratto di servizio per Windows Azure, di fatto, richiede l'esecuzione di almeno due istanze di ogni ruolo. Le applicazioni che eseguono una sola istanza di qualsiasi ruolo non possono ottenere le garanzie previste da questo contratto di servizio.

Il modo più comune per raggiungere questo obiettivo è rendere ogni istanza di un ruolo equivalente, come nel caso dei ruoli Web con bilanciamento del carico che accettano le richieste degli utenti. Questo, comunque, non è strettamente necessario, purché il malfunzionamento di una singola istanza del ruolo non determini l'arresto dell'applicazione. Ad esempio, un'applicazione potrebbe utilizzare un gruppo di istanze di un ruolo di lavoro, ognuna delle quali include dati diversi, per memorizzare nella cache i dati per le istanze del ruolo Web. Se un'istanza del ruolo di lavoro presenta problemi, un'istanza di un ruolo Web che tenti di accedere ai dati memorizzati nella cache contenuti nella prima istanza si comporterà come se non riuscisse a individuare i dati nella cache (ad esempio, accederà all'archiviazione permanente per risolvere il problema). Il problema potrebbe causare un rallentamento dell'esecuzione dell'applicazione che tuttavia, dal punto di vista di un utente, continuerà a presentare un comportamento corretto.

Un altro punto importante da tenere presente è che, anche se l'applicazione di esempio illustrata finora include solo ruoli Web e di lavoro, tutte queste regole si applicano anche alle applicazioni che utilizzano i ruoli VM. Esattamente come gli altri, è necessario che ogni ruolo VM esegua almeno due istanze per qualificarsi per il contratto di servizio di Windows Azure e che l'applicazione continui a funzionare correttamente se una di esse presenta un problema. Anche con i ruoli VM, Windows Azure continua a offrire una soluzione di tipo PaaS, non un sistema IaaS tradizionale.

CHE COSA OFFRE IL MODELLO DI PROGRAMMAZIONE WINDOWS AZURE

Il modello di programmazione Windows Azure è basato su Windows e la maggior parte delle competenze di uno sviluppatore Windows è applicabile a questo nuovo ambiente. Nonostante questo, il nuovo modello è diverso dal modello di programmazione Windows Server convenzionale. Perché, allora, vale la pena di apprenderlo? In che modo può facilitare la creazione di applicazioni più efficienti? Per rispondere a queste domande, è innanzitutto necessario esaminare meglio il funzionamento di Windows Azure. Una volta chiarito questo aspetto, diventerà semplice comprendere in che modo il modello di programmazione Windows Azure può facilitare la creazione di software migliore.

CONCETTI DI BASE: IL CONTROLLER DI INFRASTRUTTURA

Windows Azure è progettato per l'esecuzione all'interno di data center che includono un numero elevato di computer. Di conseguenza, ogni applicazione Windows Azure viene eseguita in più computer contemporaneamente. Nella figura 4 questo concetto è illustrato con un semplice esempio.

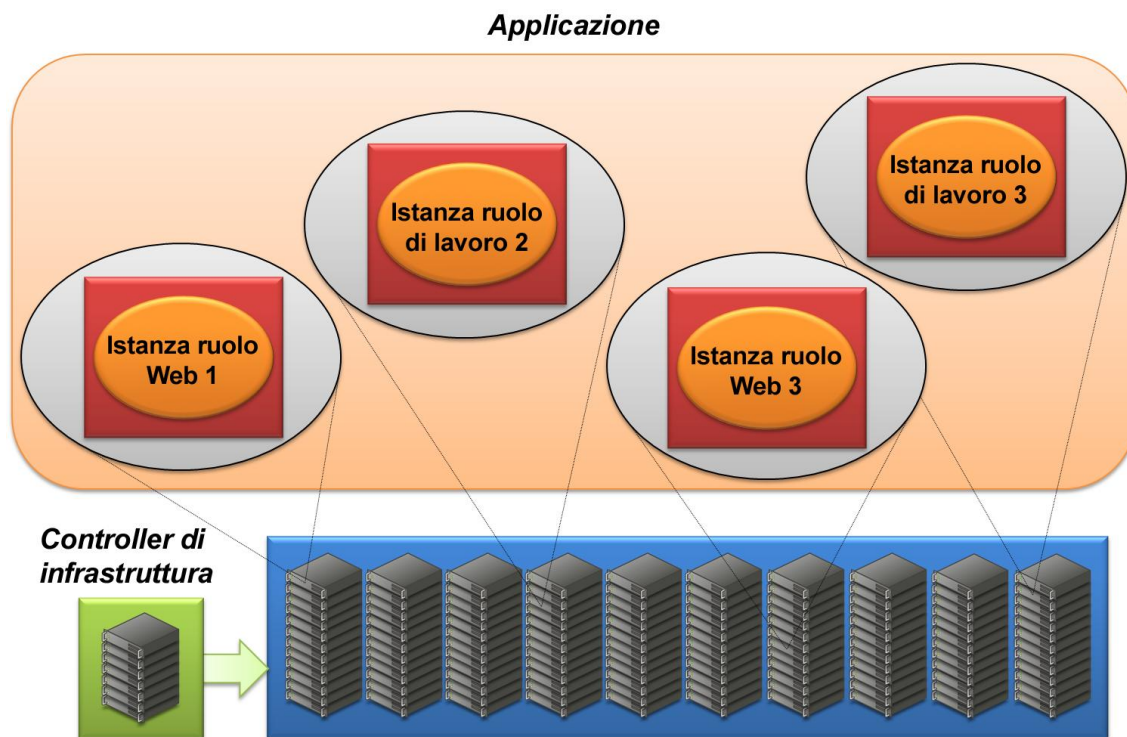


Figura 4: il controller di infrastruttura di Windows Azure crea istanze dei ruoli di un'applicazione in diversi computer, quindi ne controlla l'esecuzione.

Come illustrato nella figura 4, tutti i computer in un determinato data center Windows Azure sono gestiti da un'applicazione denominata *controller di infrastruttura*. Lo stesso controller di infrastruttura è un'applicazione distribuita eseguita in più computer.

Per eseguire un'applicazione in Windows Azure lo sviluppatore deve specificare il codice per i ruoli dell'applicazione, insieme ai relativi file di definizione e di configurazione del servizio. Tra le altre cose, queste informazioni indicano al controller di infrastruttura quante istanze creare per ogni ruolo. Il controller di infrastruttura seleziona una macchina fisica per ogni istanza, crea su di essa una macchina virtuale e avvia l'esecuzione dell'istanza. Come suggerisce la figura, le istanze dei ruoli per una singola applicazione sono suddivise tra i vari computer all'interno del data center.

Una volta create, le istanze vengono monitorate costantemente dal controller di infrastruttura. Se un'istanza non funziona per qualsiasi motivo, hardware o software, il controller avvia una nuova istanza per il relativo ruolo. Anche se eventuali malfunzionamenti possono provocare una temporanea riduzione del numero delle istanze di un'applicazione al di sotto della soglia richiesta dallo sviluppatore, il controller di infrastruttura avvierà sempre tante nuove istanze quante sono necessarie per mantenere il numero richiesto per ogni ruolo dell'applicazione. Nella figura 4 sono illustrati solo ruoli Web e ruoli di lavoro, tuttavia anche i ruoli VM vengono gestiti nello stesso modo e ognuna delle istanze del ruolo viene eseguita in un computer fisico diverso.

I VANTAGGI: AMMINISTRAZIONE, DISPONIBILITÀ E SCALABILITÀ MIGLIORATE

Le applicazioni create con il modello di programmazione Windows Azure possono risultare più semplici da amministrare, più disponibili e più scalabili rispetto a quelle basate sui server Windows tradizionali. Vale la pena di esaminare separatamente queste tre caratteristiche.

I vantaggi di Windows Azure in termini di amministrazione derivano in gran parte dal controller di infrastruttura. Come qualsiasi altro sistema operativo, Windows richiede l'applicazione di patch. Negli ambienti locali questo tipo di attività richiede un intervento da parte di un operatore. In Windows Azure, invece, il processo è completamente automatizzato: il controller di infrastruttura gestisce gli aggiornamenti per le istanze dei ruoli Web e di lavoro (ma non per le istanze dei ruoli VM). Quando necessario, la piattaforma esegue anche l'aggiornamento dei server Windows sottostanti, in cui sono in esecuzione le macchine virtuali. Il risultato è una riduzione dei costi, dal momento che per gestire questa funzione non sono necessari amministratori.

Naturalmente, ridurre i costi mediante una riduzione delle attività di amministrazione è positivo, così come garantire una maggiore disponibilità delle applicazioni. Il modello di programmazione Windows Azure aumenta la disponibilità delle applicazioni in vari modi, ovvero:

- *Protezione contro i problemi hardware. Dato che ogni applicazione è costituita da più istanze di ogni ruolo, i problemi hardware, come il guasto di un disco, un errore di rete o l'arresto anomalo di un computer server, non determinano l'arresto dell'applicazione. A questo scopo, il controller di infrastruttura non seleziona in modo casuale i computer per le istanze di un'applicazione. Istanze diverse dello stesso ruolo vengono invece collocate in diversi domini di errore. Un dominio di errore è un insieme di componenti hardware (computer, switch e di altro tipo) che condividono un singolo punto di errore. Ad esempio, tutti i computer in un singolo dominio di errore potrebbero utilizzare lo stesso switch per connettersi alla rete. Per questo motivo, un singolo problema hardware non determina l'arresto*

dell'intera applicazione. È possibile che alcune istanze vadano perdute, ma l'applicazione continuerà comunque a comportarsi correttamente.

- *Protezione contro i malfunzionamenti software. Oltre ai problemi hardware, il controller di infrastruttura è in grado di rilevare anche malfunzionamenti causati dal software. Se il codice in un'istanza causa un arresto anomalo o la macchina virtuale in cui è in esecuzione non è più disponibile, il controller di infrastruttura avvierà solo il codice oppure, se necessario, una nuova macchina virtuale per tale ruolo. Anche se l'eventuale attività in corso nell'istanza al momento dell'interruzione andrà persa, la nuova istanza verrà immediatamente creata all'avvio dell'esecuzione dell'applicazione.*
- *Capacità di aggiornare le applicazioni senza tempi di inattività. Tutte le applicazioni richiedono aggiornamenti, che si tratti di interventi di manutenzione ordinaria o dell'installazione di una nuova versione. Un'applicazione creata utilizzando il modello di programmazione Windows Azure può essere aggiornata mentre è in esecuzione, senza necessità di arrestarla. A questo scopo, istanze diverse per ciascuno dei ruoli di un'applicazione vengono collocate in diversi domini di aggiornamento (diversi dai domini di errore descritti sopra). Quando è necessario distribuire una nuova versione dell'applicazione, il controller di infrastruttura è in grado di arrestare le istanze in un solo dominio di aggiornamento, aggiornare il codice e quindi creare nuove istanze dal nuovo codice. Una volta che tali istanze sono in esecuzione, è possibile ripetere le stesse operazioni per le istanze nel dominio di aggiornamento successivo e così via. Durante questo processo, gli utenti potranno visualizzare versioni diverse delle applicazioni, a seconda dell'istanza con la quale interagiscono, ma l'applicazione nel complesso rimarrà costantemente disponibile.*
- *Capacità di aggiornare Windows e altro software di supporto senza tempi di inattività delle applicazioni. Il controller di infrastruttura presuppone che ogni applicazione Windows Azure rispetti le tre regole illustrate in precedenza. Questo implica la possibilità di arrestare alcune delle istanze di un'applicazione in qualsiasi momento, aggiornare il software di sistema sottostante e quindi avviare nuove istanze. Eseguendo queste operazioni in blocchi, senza mai arrestare contemporaneamente tutte le istanze di un ruolo, è possibile aggiornare Windows e altri software mentre l'applicazione resta costantemente in esecuzione.*

La disponibilità è essenziale per la maggior parte delle applicazioni (il software non è funzionale se non è in esecuzione quando l'utente ne ha bisogno), ma anche la scalabilità ha la sua importanza. Il modello di programmazione Windows Azure consente agli sviluppatori di creare applicazioni più scalabili essenzialmente in due modi:

- *Tramite la creazione e la gestione automatiche di un numero specificato di istanze di ruolo. Come già descritto, lo sviluppatore indica a Windows Azure il numero di istanze di ogni ruolo da eseguire e il controller di infrastruttura crea e monitora le istanze richieste. Questo rende la scalabilità dell'applicazione estremamente lineare: è sufficiente che lo sviluppatore indichi a Windows Azure quali sono le sue esigenze. Dato che questa piattaforma cloud viene eseguita in data center di grandi dimensioni, ottenere il livello di scalabilità richiesto da un'applicazione, qualunque esso sia, generalmente non è un problema.*
- *Offrendo un sistema per modificare il numero di istanze di ruolo in esecuzione per un'applicazione attiva. Per le applicazioni il cui carico di lavoro varia, ottenere la scalabilità è più complicato. Impostare il numero di istanze una volta per tutte non è una buona soluzione, in quanto carichi diversi possono determinare un aumento o una riduzione significativa del numero ideale di istanze. Per gestire questa situazione, Windows Azure fornisce sia un portale Web per gli utenti che un'API per le applicazioni, per*

consentire di modificare il numero desiderato di istanze per ogni ruolo, mentre l'applicazione è in esecuzione.

Semplificare l'amministrazione delle applicazioni e renderle più disponibili e più scalabili è sicuramente utile ed è per questo che utilizzare il modello di programmazione Windows Azure è in generale una scelta valida. Tuttavia, come accennato in precedenza, è possibile eseguire in Windows Azure anche applicazioni che non seguono questo modello. Si supponga, ad esempio, di creare un'applicazione che utilizza un unico ruolo (il che è consentito), ma di eseguire poi una sola istanza di tale ruolo (il che viola la seconda e la terza regola). Questa scelta potrebbe essere dettata da considerazioni economiche, dal momento che in Windows Azure è previsto un addebito per ogni istanza in esecuzione. È tuttavia importante tenere presente che, in questo caso, il controller di infrastruttura non può sapere che la relativa applicazione non segue tutte e tre le regole: il controller arresterà la singola istanza in modo imprevedibile, per applicare eventuali patch al software sottostante, e quindi riavvierà una nuova istanza. Per gli utenti, questo significa che l'applicazione risulterà occasionalmente non disponibile, dal momento che non sono presenti altre istanze che possono sostituire la prima. Questo non è un bug di Windows Azure, ma un aspetto fondamentale della modalità di funzionamento della tecnologia.

Per ottenere tutti i vantaggi offerti da Windows Azure è necessario conformarsi alle regole del relativo modello di programmazione. Trasferire applicazioni esistenti da Windows Server a Windows Azure può richiedere un certo impegno, come descritto in dettaglio più avanti in questo documento. Per le nuove applicazioni, tuttavia, i punti a favore dell'utilizzo del modello Windows Azure sono evidenti. Perché non creare un'applicazione meno costosa da amministrare, costantemente disponibile e facilmente scalabile sia verso l'alto che verso il basso? In futuro, è ragionevole aspettarsi che sempre più applicazioni verranno create utilizzando il modello di programmazione Windows Azure.

IMPLICAZIONI DEL MODELLO DI PROGRAMMAZIONE WINDOWS AZURE: ALTRE MODIFICHE

Creare applicazioni per Windows Azure implica il rispetto delle tre regole del relativo modello di programmazione. Questo, tuttavia, non è sufficiente: esistono anche altri ambiti dell'attività di sviluppo che richiedono adattamenti. Le modifiche apportate dal modello di programmazione Windows Azure all'ambiente di sviluppo più in generale possono essere raggruppate in tre aree:

- *La modalità di interazione delle istanze di ruolo con il sistema operativo.*
- *La modalità di interazione delle istanze di ruolo con l'archiviazione permanente.*
- *La modalità di interazione delle istanze di ruolo tra loro.*

In questa sezione vengono esaminate tutte e tre le modalità.

INTERAZIONI CON IL SISTEMA OPERATIVO

Tipicamente, un'applicazione in esecuzione in un computer Windows Server è sotto il controllo del relativo amministratore, che può riavviare le macchine virtuali o il computer in cui sono in esecuzione, installare patch di Windows ed eseguire qualsiasi altra operazione necessaria per mantenere la disponibilità del computer. In Windows Azure, invece, tutti i server sono di proprietà del controller di infrastruttura, che decide quando riavviare le macchine virtuali o i computer e, per i ruoli Web e di lavoro

(ma non per i ruoli VM), esegue l'installazione delle patch e degli altri aggiornamenti del software di sistema in ogni istanza.

Questo approccio presenta vantaggi effettivi, come è stato descritto, ma determina anche alcune limitazioni. Dato che il controller di infrastruttura detiene il controllo delle macchine fisiche e virtuali utilizzate dalle applicazioni Windows Azure, può eseguire su di esse qualsiasi operazione. Di conseguenza, lasciare che un'applicazione Windows Azure modifichi il sistema in cui è in esecuzione, ovvero lasciare che venga eseguita in modalità amministratore piuttosto che in modalità utente, presenta alcune problematiche. Dato che il controller di infrastruttura può modificare il sistema operativo a proprio piacimento, non esiste alcuna garanzia che le modifiche apportate da un'istanza di ruolo al sistema in cui è in esecuzione non vengano sovrascritte. Inoltre, le specifiche macchine virtuali (e fisiche) in cui un'applicazione viene eseguita cambiano nel tempo. Questo implica che le eventuali modifiche apportate all'ambiente locale predefinito dovranno essere ripetute a ogni avvio di un'istanza di ruolo.

Nella sua prima versione, Windows Azure semplicemente non consentiva alle applicazioni di modificare i sistemi nei quali venivano eseguite: in altre parole, le applicazioni venivano eseguite solo in modalità utente. Questa limitazione è stata attenuata: tramite i ruoli Web e di lavoro gli sviluppatori oggi possono eseguire le applicazioni in modalità amministratore, ma il modello di programmazione nel complesso non è cambiato. Chiunque crei un'applicazione Windows Azure deve comprendere il funzionamento del controller di infrastruttura e progettare le applicazioni di conseguenza.

INTERAZIONI CON L'ARCHIVIAZIONE PERMANENTE

Le applicazioni non sono costituite solo da codice, ma utilizzano anche dati. Così come il modello di programmazione deve evolvere per rendere le applicazioni più disponibili e più scalabili, anche la modalità di archiviazione e di accesso ai dati deve cambiare. Le modifiche principali sono le seguenti:

- *L'archiviazione deve essere esterna alle istanze di ruolo. Anche se ogni istanza dispone di una macchina virtuale con un proprio file system, i dati archiviati in questi file system non vengono automaticamente resi permanenti. In caso di problemi di un'istanza, tutti i dati in essa contenuti possono andare persi. Di conseguenza, perché le applicazioni funzionino correttamente in caso di problemi, è necessario che i dati siano archiviati in modo permanente al di fuori delle istanze di ruolo. In questo modo i dati, che andrebbero perduti se archiviati in locale in un'istanza che genera problemi, saranno accessibili da un'altra istanza.*
- *È necessario che l'archiviazione sia replicata. Così come un'applicazione Windows Azure esegue più istanze di ruolo per gestire eventuali errori, anche l'archiviazione Windows Azure deve prevedere più copie dei dati. In caso contrario, un singolo problema sarebbe sufficiente a rendere i dati non disponibili: un evento inaccettabile per le applicazioni ad alta disponibilità.*
- *È necessario che l'archiviazione gestisca quantità elevate di dati. I sistemi relazionali tradizionali non sono necessariamente la scelta migliore per i set di dati di grandi dimensioni. Dato che Windows Azure è progettato in parte per applicazioni altamente scalabili, deve fornire meccanismi di archiviazione per la gestione di dati a tale livello. A questo scopo, la piattaforma offre blob per l'archiviazione di dati binari insieme a un approccio non SQL costituito da tabelle per l'archiviazione di grandi set di dati strutturati.*

Nella figura 5 sono descritte queste tre caratteristiche, illustrando come un'applicazione viene considerata dal punto di vista dell'archiviazione in Windows Azure.

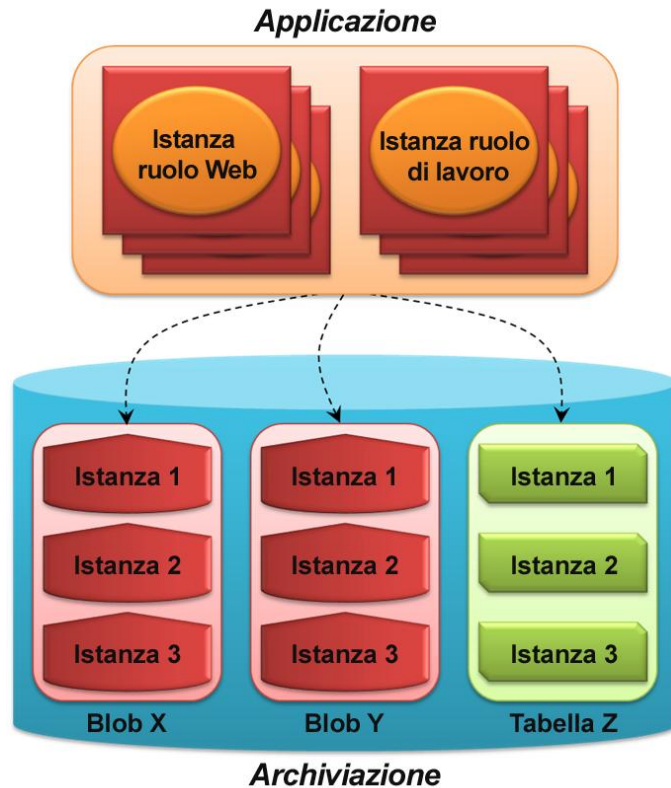


Figura 5: anche se le applicazioni vedono una singola copia, l'archiviazione Windows Azure replica tutti i blob e le tabelle tre volte.

In questo esempio, un'applicazione Windows Azure utilizza due blob e una tabella dall'archiviazione Windows Azure. L'applicazione vede i blob e la tabella come singole entità ma, in realtà, l'archiviazione Windows Azure mantiene tre istanze di ogni elemento. Le copie sono distribuite in diversi computer fisici che, come avviene per le istanze di ruolo, si trovano in domini di errore diversi. Questo migliora la disponibilità dell'applicazione, dal momento che i dati restano accessibili anche quando una o più copie non sono disponibili. Inoltre, poiché i dati permanenti vengono archiviati esternamente alle istanze di ruolo dell'applicazione, in caso di errore di un'istanza andranno persi solo i dati in uso quando si è verificato il problema.

Il modello di programmazione Windows Azure richiede che un'applicazione presenti un comportamento corretto in caso di problemi di un'istanza di ruolo. A questo scopo, ogni istanza di un'applicazione deve archiviare tutti i dati permanenti nell'archiviazione Windows Azure o in un altro meccanismo di archiviazione esterno (come SQL Azure, il servizio di Microsoft basato sulla cloud per i dati relazionali). Esiste però un'ulteriore opzione degna di nota: le unità Windows Azure. Come già accennato, i dati scritti da un'applicazione nel file system locale della relativa macchina virtuale possono andare perduti in caso di arresto anomalo della macchina virtuale. Con le unità Windows Azure ciò non avviene, in quanto queste utilizzano un blob per offrire archiviazione permanente per il file system di una particolare istanza. Queste unità possono essere utili in determinate situazioni, anche se presentano alcune limitazioni: ad esempio, solo un'istanza per volta può leggere e scrivere su una particolare unità Windows Azure, mentre a tutte le altre istanze nell'applicazione è consentito solo l'accesso in lettura.

INTERAZIONI TRA LE ISTANZE DI RUOLO

Quando un'applicazione è suddivisa in più parti, normalmente è necessaria una certa interazione tra di esse. In un'applicazione Windows Azure, questo avviene tramite la comunicazione tra le istanze di ruolo. Ad esempio, un'istanza di ruolo Web può accettare le richieste degli utenti e quindi trasferirle a un'istanza di ruolo di lavoro, a scopo di ulteriore elaborazione.

Questa interazione non avviene come nelle normali applicazioni Windows. Ancora una volta, un elemento fondamentale da ricordare è che, nella maggior parte dei casi, tutte le istanze di un determinato ruolo sono equivalenti e, quindi, intercambiabili. Di conseguenza quando, ad esempio, un'istanza di ruolo Web trasferisce un'attività a un'istanza di ruolo di lavoro, non è importante quale sia l'istanza di destinazione. Di fatto, l'istanza di ruolo Web non dovrebbe fare affidamento, per comunicare con un'altra istanza, su elementi specifici dell'istanza, come l'indirizzo IP di un'istanza di ruolo di lavoro. Sono necessari meccanismi più generici.

La modalità più comune per la comunicazione tra le istanze di ruolo nelle applicazioni Windows Azure è rappresentata dalle code Windows Azure. Nella figura 6 viene illustrato questo concetto.

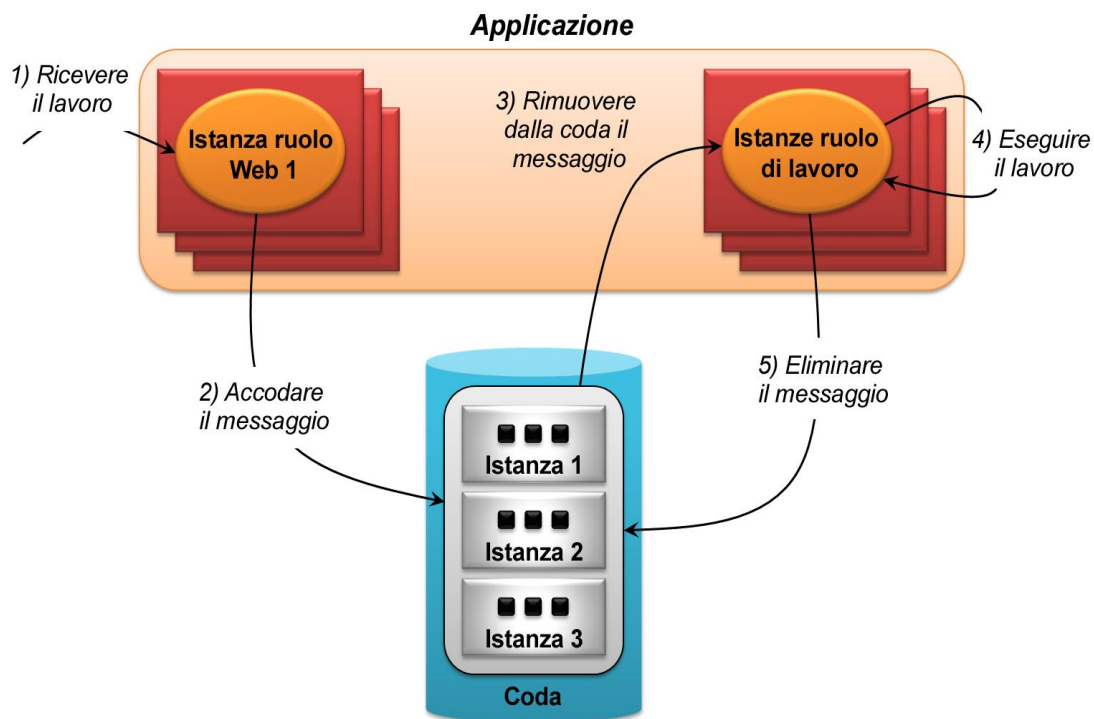


Figura 6: le istanze di ruolo possono comunicare attraverso le code, ognuna delle quali replica tre volte i messaggi in essa contenuti.

Nell'esempio, un'istanza di ruolo Web ottiene un'attività da eseguire da un utente dell'applicazione, ad esempio una persona che esegue una richiesta tramite un browser (fase 1). Quindi, l'istanza crea un messaggio contenente l'attività e lo scrive in una coda Windows Azure (fase 2). Queste code sono implementate come parte dell'archiviazione Windows Azure; di conseguenza, come per blob e tabelle, ogni coda viene replicata tre volte, come indicato nella figura. Come di consueto, questo implica tolleranza di errore, assicurando che i messaggi della coda restino disponibili in caso di problemi.

Successivamente, un'istanza di ruolo di lavoro legge il messaggio dalla coda (fase 3). Si noti che, per l'istanza di ruolo Web che ha creato il messaggio, non ha importanza quale sia l'istanza di ruolo di lavoro di destinazione: in questa applicazione sono tutte equivalenti. L'istanza di ruolo di lavoro svolge l'attività richiesta dal messaggio (fase 4) e quindi lo elimina dalla coda (fase 5).

Quest'ultima fase, ovvero la rimozione esplicita del messaggio dalla coda, funziona diversamente rispetto alle tradizionali tecnologie di accodamento locali. In Accodamento messaggi Microsoft (MicrosoftMessageQueuing), ad esempio, un'applicazione può eseguire una lettura all'interno di una transazione atomica. Se l'applicazione presenta un problema prima di completare l'attività prevista, la transazione si interrompe e il messaggio viene automaticamente reinserito in coda. Questo garantisce che ogni messaggio inviato a una coda MSMQ venga recapitato solo una volta e nell'ordine in cui è stato inviato.

Le code Windows Azure non supportano le letture transazionali e, di conseguenza, non garantiscono che il recapito avvenga una sola volta e nell'ordine di invio. Nell'esempio della figura 6, può accadere che l'istanza di ruolo di lavoro termini l'elaborazione del messaggio e quindi si interrompa appena prima di eliminare il messaggio dalla coda. In tal caso, il messaggio riapparirà automaticamente dopo un periodo di timeout configurabile e verrà elaborato da un'altra istanza di ruolo di lavoro. A differenza di quanto avviene in MSMQ, le code Windows Azure includono una semantica di tipo "at-least-once": un messaggio può essere letto ed elaborato una o più volte.

Questo solleva una domanda ovvia: perché le code Windows Azure non supportano le letture transazionali? La risposta è che le transazioni richiedono meccanismi di blocco, e questo implica necessariamente un rallentamento delle operazioni (soprattutto tenuto conto della replica dei messaggi prevista dalle code Windows Azure). Considerati gli obiettivi principali della piattaforma, i suoi progettisti hanno optato per l'approccio più veloce e scalabile.

Nella maggior parte dei casi, le code rappresentano la modalità di comunicazione ottimale tra le istanze di ruolo all'interno di un'applicazione. Le istanze, tuttavia, possono anche interagire direttamente, senza passare attraverso una coda. A questo scopo, Windows Azure offre un'API che consente a un'istanza di individuare tutte le altre istanze della stessa applicazione che soddisfano determinati requisiti e quindi di inviare una richiesta direttamente a una di esse. Nel caso più comune, in cui tutte le istanze di un determinato ruolo sono equivalenti, l'istanza che esegue la chiamata dovrebbe scegliere in modo casuale un'istanza di destinazione dal set restituito dall'API. Questo non è sempre vero: è possibile che un ruolo di lavoro implementi una cache in memoria con ogni istanza di ruolo contenente dati specifici e quindi che l'istanza che esegue la chiamata debba accedere a una particolare istanza. Nella maggior parte dei casi, tuttavia, l'approccio corretto consiste nel gestire tutte le istanze di un ruolo come intercambiabili.

TRASFERIMENTO DI APPLICAZIONI WINDOWS SERVER IN WINDOWS AZURE

Chiunque crei una nuova applicazione Windows Azure dovrebbe seguire le tre regole del relativo modello di programmazione. Per essere trasferita in Windows Azure, tuttavia, anche un'applicazione Windows Server esistente dovrà seguire le stesse regole. In aggiunta, potrebbe essere necessario modificare il modo in cui l'applicazione interagisce con il sistema operativo e utilizza l'archiviazione permanente, nonché le modalità di interazione tra i suoi componenti.

La maggiore o minore facilità di queste modifiche dipende dall'applicazione. Ecco alcuni esempi rappresentativi:

- *Applicazione ASP.NET con più istanze a carico bilanciato che condividono lo stato archiviato in SQL Server. Per questo tipo di applicazione, in genere, il porting a Windows Azure avviene con facilità: ogni istanza dell'applicazione originale diventa un'istanza di ruolo Web o di lavoro. Le applicazioni come questa non utilizzano sessioni permanenti, il che contribuisce a renderle particolarmente adatte per Windows Azure (l'utilizzo dello stato della sessione ASP.NET è accettabile dato che Windows Azure offre un'opzione per archiviare lo stato della sessione in modo permanente nella tabella di archiviazione Windows Azure). Inoltre, il trasferimento di un database SQL Server locale in SQL Azure è in genere piuttosto semplice.*
- *Applicazione ASP.NET con più istanze che mantiene lo stato per ogni singola istanza e si basa su sessioni permanenti. Poiché mantiene lo stato specifico del client in ogni istanza tra una richiesta e l'altra, questa applicazione richiederà alcune modifiche. Windows Azure non supporta le sessioni permanenti. Di conseguenza, per l'esecuzione dell'applicazione su questa piattaforma cloud sarà necessario riprogettarne la modalità di gestione dello stato.*
- *Client Silverlight o Windows Presentation Foundation (WPF) che accede a servizi WCF in esecuzione su un livello intermedio. Se i servizi non mantengono lo stato per il singolo client tra le chiamate, il loro trasferimento in Windows Azure non crea particolari difficoltà. Il client continuerà a funzionare sui desktop degli utenti, come sempre, ma eseguirà le chiamate a servizi in esecuzione in Windows Azure. Se, al contrario, i servizi correnti mantengono lo stato per il singolo client, sarà necessario riprogettarli.*
- *Applicazione con una singola istanza in esecuzione in Windows Server che mantiene lo stato sul proprio computer. Che i client siano browser o di altro tipo, numerose applicazioni enterprise oggi sono progettate in questo modo e per funzionare correttamente in Windows Azure richiederanno interventi di riprogettazione. Potrebbe essere possibile eseguire questo tipo di applicazione senza modifiche in una singola istanza di ruolo VM, ma probabilmente gli utenti non saranno molto soddisfatti dei risultati. Anzitutto, il contratto di servizio di Windows Azure non si applica alle applicazioni con una sola istanza. Si tenga inoltre presente che il controller di infrastruttura può riavviare in qualsiasi momento il computer in cui è in esecuzione l'istanza per aggiornarne il software. L'applicazione non dispone di alcun controllo sul momento in cui viene eseguita l'operazione, che potrebbe avvenire in coincidenza con il picco di attività di una giornata lavorativa. In assenza di una seconda istanza in grado di subentrare alla prima (poiché l'applicazione non è stata creata in conformità alle regole del modello di programmazione Windows Azure), l'applicazione non sarà disponibile per un certo periodo di tempo. Di conseguenza, chiunque la utilizzi subirà un'interruzione del lavoro mentre il computer viene riavviato. Anche se il ruolo VM facilita il trasferimento di un file binario di Windows Server in Windows Azure, questo non garantisce di per sé che l'applicazione funzionerà correttamente nella nuova piattaforma. È inoltre necessario che l'applicazione sia conforme alle tre regole del modello di programmazione Windows Azure.*
- *Applicazione Visual Basic 6 che accede direttamente a un database SQL Server, ovvero un'applicazione client/server tradizionale. Il corretto funzionamento di questa applicazione in Windows Azure richiederà probabilmente almeno la riscrittura della logica di business del client. Anche se potrebbe essere possibile trasferire il database (includere eventuali stored procedure) in SQL Azure e quindi reindirizzare i client alla nuova posizione, il componente desktop dell'applicazione non funzionerà senza modifiche in Windows Azure. Windows Azure non include un'interfaccia utente locale e inoltre non supporta l'utilizzo di Servizi Desktop in remoto (in precedenza denominato Servizi terminal) per rendere disponibili interfacce utente remote.*

Windows Azure aiuta gli sviluppatori a realizzare applicazioni più efficienti. Tuttavia, i miglioramenti che offre richiedono determinate modifiche. Di conseguenza, trasferire il software esistente su questa nuova piattaforma può richiedere un certo impegno. Per prendere la decisione giusta è necessario comprendere sia il potenziale valore di business che le eventuali problematiche tecniche collegate al trasferimento di un'applicazione in Windows Azure.

CONCLUSIONI

Le piattaforme cloud rappresentano un nuovo universo, che apre nuove opportunità. In linea con questa evoluzione, il modello di programmazione Windows Azure consente agli sviluppatori di creare applicazioni più semplici da amministrare, più disponibili e più scalabili rispetto a quelle realizzate nel tradizionale ambiente Windows Server. A questo scopo è necessario seguire tre regole:

- *Un'applicazione Windows Azure è costituita da uno o più ruoli.*
- *Un'applicazione Windows Azure esegue più istanze di ciascun ruolo.*
- *Un'applicazione Windows Azure presenta un comportamento corretto in caso di problemi di un'istanza di un ruolo.*

Per utilizzare con successo questo modello di programmazione è inoltre necessario comprendere i cambiamenti che esso apporta alle modalità con cui le applicazioni interagiscono con il sistema operativo, utilizzano l'archiviazione permanente e rendono possibile la comunicazione tra le istanze di ruolo. Per gli sviluppatori disposti a impegnarsi in questo senso, comunque, i vantaggi sono evidenti. Sebbene non rappresenti la scelta ideale per ogni scenario, il modello di programmazione Windows Azure può risultare utile per chiunque desideri creare applicazioni più semplici da amministrare, disponibili e scalabili.

PER ULTERIORI INFORMAZIONI

Introduzione a Windows Azure (in inglese): <http://go.microsoft.com/?linkid=9682907>

Introduzione alla piattaforma Windows Azure (in inglese):
<http://go.microsoft.com/fwlink/?LinkId=158011>

INFORMAZIONI SULL'AUTORE

David Chappell è a capo di Chappell & Associates (www.davidchappell.com), con sede a San Francisco, in California. Attraverso la sua attività di relatore, scrittore e consulente, aiuta professionisti in tutto il mondo a comprendere, utilizzare e prendere decisioni più consapevoli sulle nuove tecnologie.