

Middleware - comunicare asincronă prin cozi de mesaje

Middleware este software-ul care conectează cererile bazate pe rețea generate de un client la datele back-end pe care clientul le solicită. Este un termen general pentru software care servește pentru a "lipi împreună" programe separate, adesea complexe și deja existente.

Programele middleware vin în software-ul local și serviciile cloud și pot fi utilizate independent sau împreună, în funcție de cazul de utilizare. Cu interacțiuni bazate pe rețea, un client sau un program solicitant poate face o solicitare. Clientul respectiv este în mod obișnuit o aplicație care se află la capătul frontal, unde interacționează utilizatorul cu software-ul. Resursele, cum ar fi bazele de date, cozile de mesaje, magazinele de date NoSQL și serverele de fișiere sunt deseori menționate ca fiind parte din back-end.

Message-oriented middleware (MOM) este o infrastructură software sau hardware care suportă trimiterea și primirea de mesaje între sistemele distribuite. MOM permite ca modulele de aplicații să fie distribuite pe platforme eterogene și reduc complexitatea dezvoltării de aplicații care acoperă mai multe sisteme de operare și protocoale de rețea. Mijloacele middleware creează un strat de comunicații distribuite care izolează dezvoltatorul de aplicații din detaliile diferitelor sisteme de operare și interfețe de rețea. API-urile care se extind pe diferite platforme și rețele sunt de obicei furnizate de MOM.

În timp ce MOM acceptă atât mesageria sincronă cât și cea asincronă, aceasta este identificată cel mai bine cu mesageria asincronă utilizând așteptarea. MOM trimite mesaje de la o aplicație la alta folosind o coadă ca etapă intermediară. Mesajele client sunt trimise către o coadă și rămân acolo până când sunt preluate de aplicația server. Avantajul acestui sistem este că aplicația de server nu trebuie să fie disponibilă atunci când mesajul este trimis, în schimb, serverul poate prelua mesajul în orice moment. În plus, deoarece mesajele pot fi extrase din coadă în orice ordine, MOM poate facilita și recuperarea mesajelor folosind scheme de prioritate sau de echilibrare a sarcinii. MOM poate oferi, de asemenea, un nivel de toleranță la erori folosind cozile persistente care permit ca mesajele să fie recuperate atunci când sistemul eșuează.

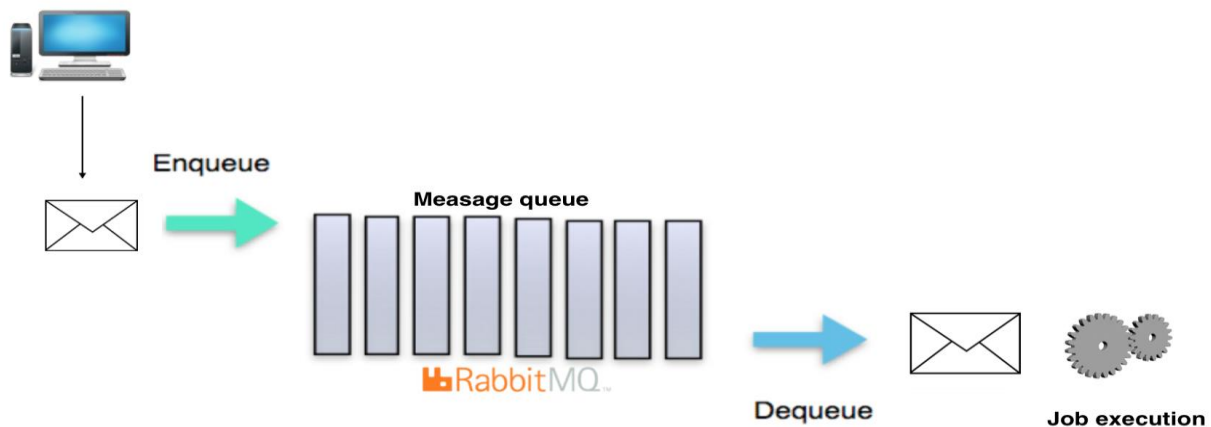
MOM furnizează elemente software care se află în toate componentele care comunică într-o arhitectură client / server și acceptă în mod tipic apeluri asincrone între aplicațiile client și server. MOM reduce implicarea dezvoltatorilor de aplicații cu complexitatea naturii master-slave a mecanismului client / server.

Mesageria asincronă în cloud este de obicei implementată utilizând cozi de mesaje, un broker de mesaje. Arhitectura de bază a unei coadă de mesaje este simplă, există aplicații client numite producători care creează mesaje și le transmit în coada de mesaje. O altă aplicație, numită consumator, se conectează la coadă și primește mesajele care urmează să fie procesate. Mesajele plasate pe coada de așteptare sunt stocate până când consumatorul le recuperează - nici măcar nu trebuie să fie difuzate simultan. Când un mesaj este preluat, acesta este procesat de către receptor

și după aceea eliminate din coadă. Acest mod de construire a unei aplicații decuplează expeditorul de la receptor - expeditorul și receptorul mesajului nu trebuie să interacționeze simultan cu coada de mesaje.

Cozile de mesaje oferă un protocol de comunicații asincron, ceea ce înseamnă că expeditorul și receptorul mesajului nu trebuie să interacționeze simultan cu coada de mesaje. Mesajele plasate pe coada sunt stocate până când destinatarul le recuperează. Cozile de mesaje au limite implicite sau explicite pentru dimensiunea datelor care pot fi transmise într-un singur mesaj și numărul de mesaje care pot rămâne nesoluționate în coadă.

Multe implementări ale cozilor de mesaje funcționează intern: într-un sistem de operare sau într-o aplicație. Alte implementări permit transmiterea de mesaje între diferite sisteme de calculatoare, care pot conecta mai multe aplicații și mai multe sisteme de operare. Aceste sisteme de așteptare a mesajelor oferă de obicei o funcționalitate îmbunătățită de rezistență pentru a se asigura că mesajele nu se "pierd" în cazul unei defecțiuni a sistemului.



Imaginați-vă că aveți un serviciu web care primește mai multe solicitări în fiecare secundă, unde nicio solicitare nu este posibilă pentru a se pierde și toate cererile trebuie procesate printr-un proces care consumă mult timp.

Imaginați-vă că serviciul web trebuie să fie întotdeauna disponibil și gata să primească o nouă solicitare, în loc să fie blocat prin procesarea cererilor primite anterior. În acest caz, este ideal să plasați o coadă între serviciul web și serviciul de procesare. Serviciul web poate pune mesajul "începe prelucrarea" pe o coadă și celălalt proces poate prelua și gestiona mesajele în ordine. Cele două procese vor fi decuplate unul de celălalt și nu trebuie să aștepte unul pe altul. Dacă aveți multe cereri venite într-o perioadă scurtă de timp, sistemul de procesare va fi capabil să le proceseze oricum. Coada de așteptare va persista în cazul în care numărul lor devine foarte mare. Puteți apoi să vă imaginați că afacerea dumneavoastră și volumul de muncă sunt în creștere și trebuie să vă scalați sistemul. Tot ce trebuie făcut acum este să adăugați mai mulți muncitori, receptoare, pentru a lucra mai rapid pe cozi.

E-mailul este probabil cel mai bun exemplu de mesagerie asincronă. Când se trimite un e-mail, expeditorul poate continua să proceseze alte lucruri fără a primi un răspuns imediat de la receptor. Acest mod de manipulare a mesajelor decuplează producătorul de la consumator. Producătorul și consumatorul mesajului nu trebuie să interacționeze simultan cu coada de mesaje.

Există adesea numeroase opțiuni în ceea ce privește semantica exactă a transmiterii mesajelor, incluzând:

- Durabilitate - mesajele pot fi păstrate în memorie, scrise pe disc sau chiar implicate într-un DBMS dacă nevoia de fiabilitate indică o soluție cu resurse mai mari.
- Politici de securitate - ce aplicații ar trebui să aibă acces la aceste mesaje?
- Politica de "curățare" a mesajelor - cozile sau mesajele pot avea un " timp pentru a trăi "
- Filtrarea mesajelor - unele sisteme acceptă filtrarea datelor astfel încât abonatul să poată vedea mesaje care corespund anumitor criterii de interes pre-specificate
- Politici de livrare - trebuie să garantăm că un mesaj este transmis cel puțin o dată sau nu?
- Politicile de rutare - într-un sistem cu multe servere de coadă, ce servere ar trebui să primească mesaje de coadă?
- Politicile de dozare - ar trebui ca mesajele să fie livrate imediat? Sau ar trebui ca sistemul să aștepte un pic și să încerce să livreze mai multe mesaje simultan?
- Criterii de așteptare - când trebuie să fie considerat un mesaj "enqueued"? Sau când a fost trimis la cel puțin o coadă de la distanță? Sau la toate cozile?
- Notificarea primirii - un editor ar putea avea nevoie să știe când unii sau toți abonații au primit un mesaj.