

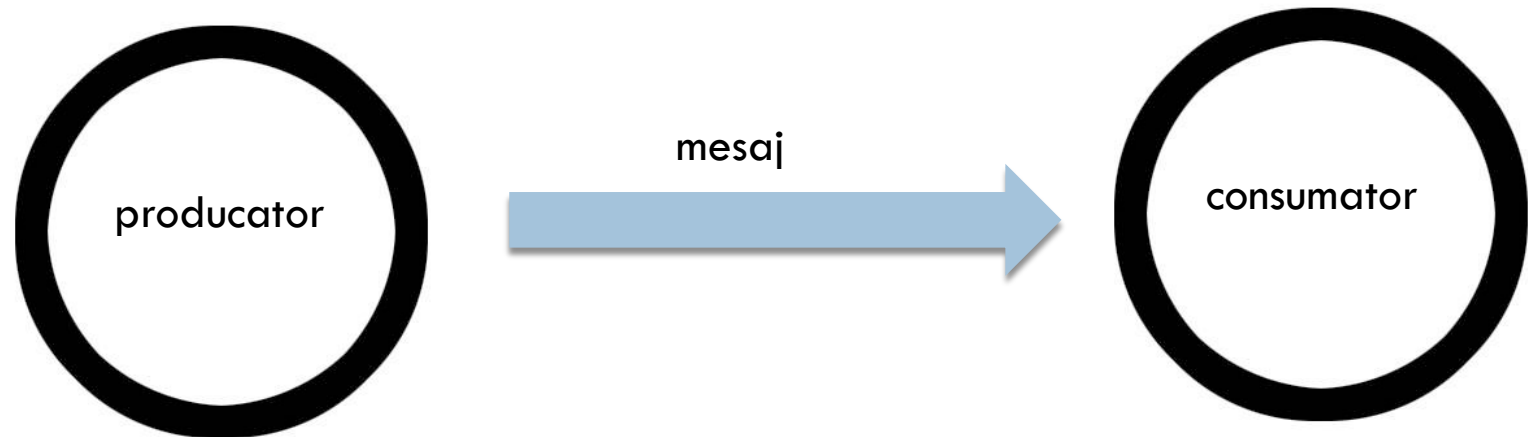
BIG DATA SI SCALABILITATE

Cozi, auto scalare si Java api pentru AWS

Cozi

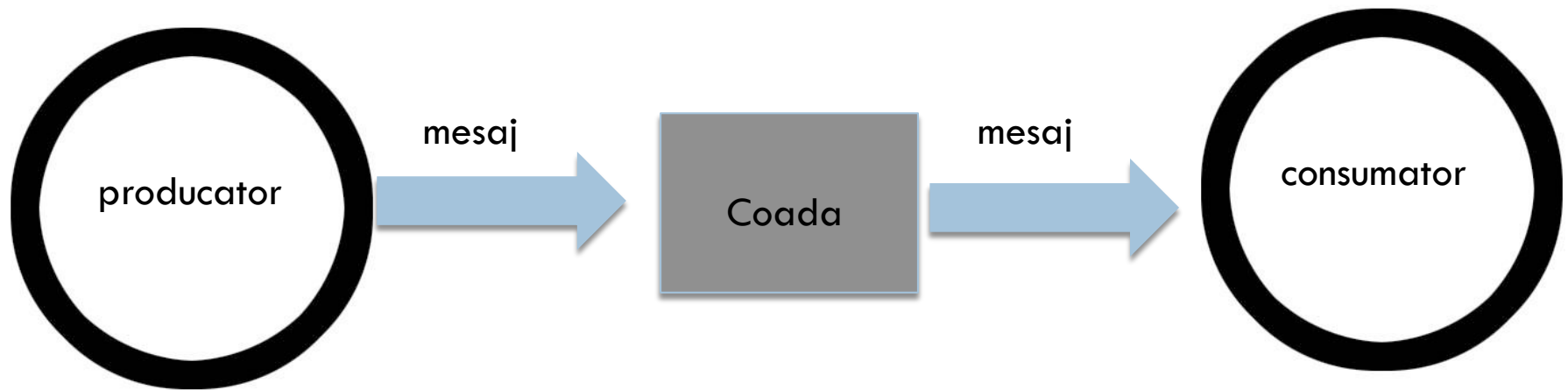


Cozi: Producator consumator



! Producatorul trimite mesaje consumatorului

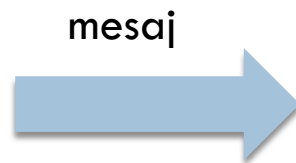
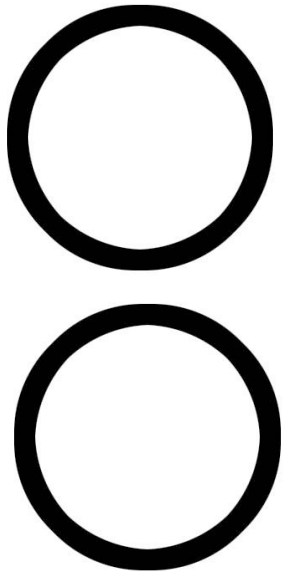
Cozi: Producator consumator



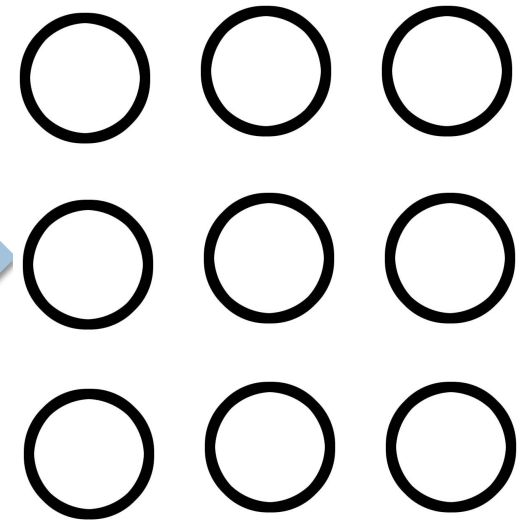
! Producatorul pune mesaje in coada, iar consumatorul extrage mesajele din coada

Cozi: Producator consumator

Producatori



Consumatori



! Numarul variabile de producatori si consumatori

Observatii

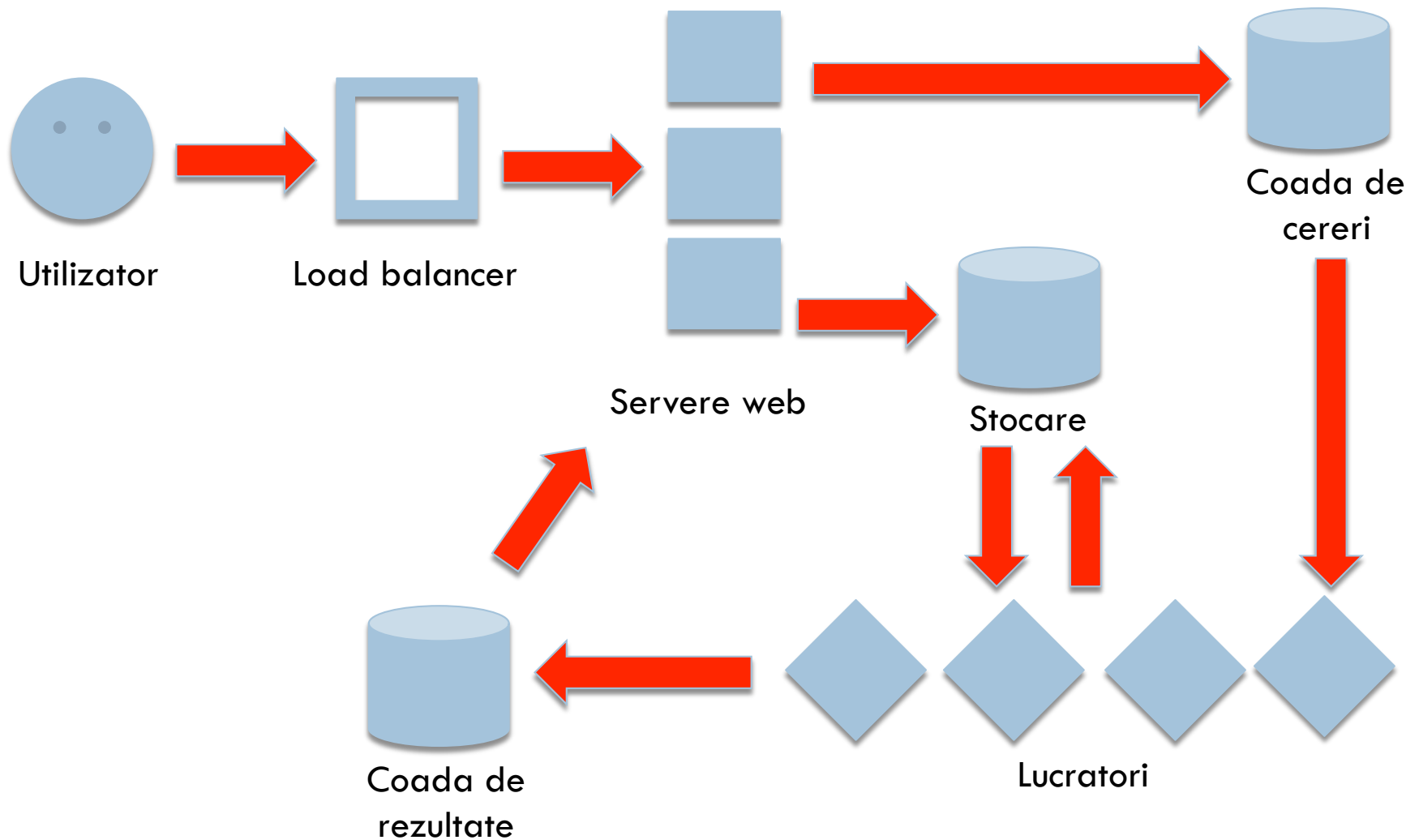
- Producatorii sunt independenti de consumatori
- Viteza cu care sunt produse mesajele in general este diferita de cea cu care sunt consumate
- Coada joaca si rol de tampon, iar consumatorii nu sunt copleșiti de cereri

Exemple de sisteme cu cozi



1. Serviciu de conversie video
2. Crawler si analizator de joburi

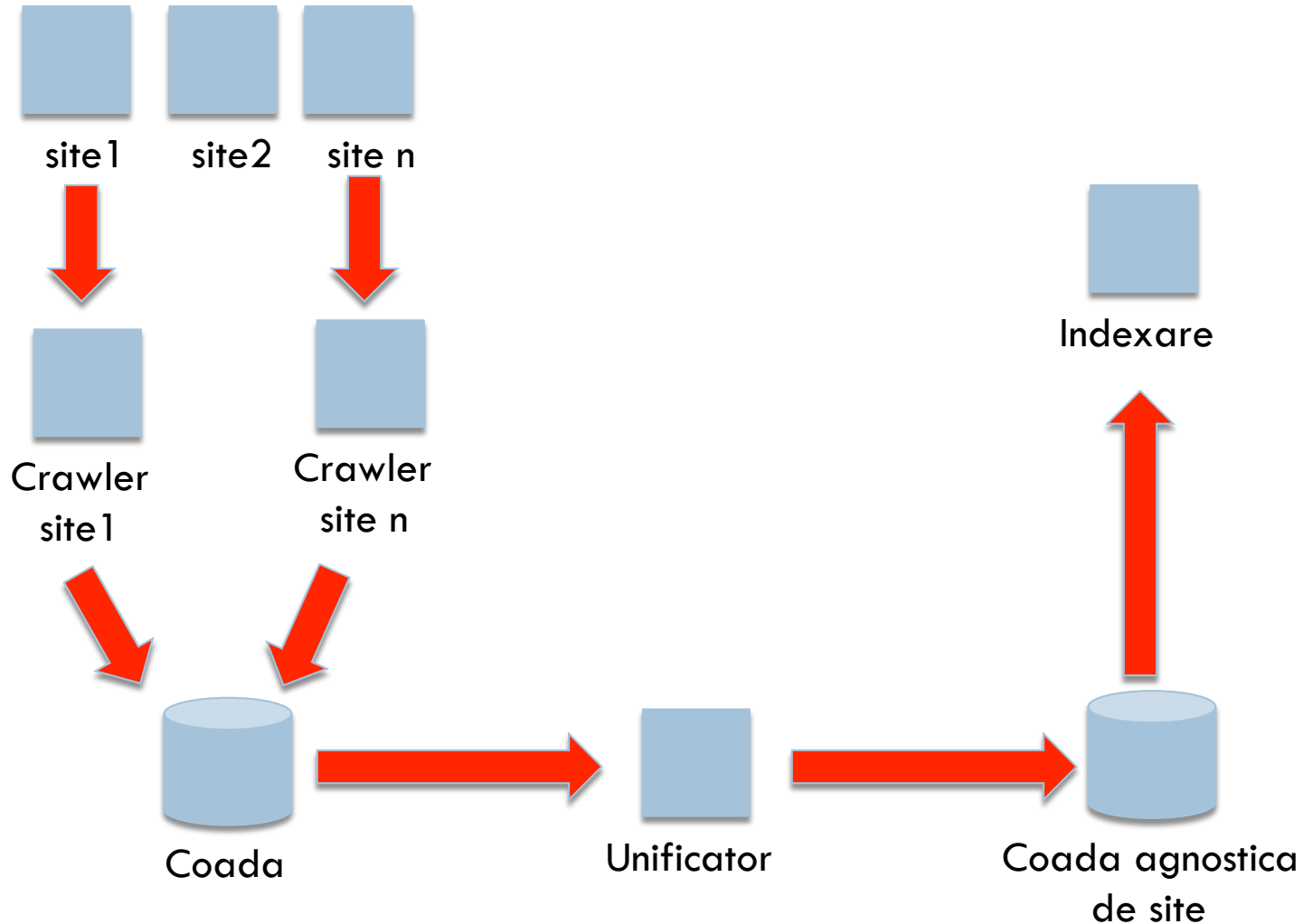
Exemplu: conversie video



Exemplu: conversie video

1. Utilizatorul cere sa se converteasca un video (trimite fisierul)
2. Serverul web pune fisierul in spatiul de stocare si un mesaj in coada de cereri
3. Un lucrator extrage mesajul. Descarca fisierul si il converteste. Pune rezultatul in spatiul de stocare si trimite un mesaj in coada de rezultate.
4. Utilizatorul cere starea procesarii serverului web. Acesta verifica coda de rezultate si raspunde.

Exemplu: crawler de job-uri



Exemplu: crawler de joburi

1. Masinile dedicate procesarii unui site rasfoiesc lista de joburi (se plimba prin linkuri). Fiecare pagina cu un job este pusa in coada (ca HTML ...)
2. O masina de unificator extrage mesajul din coada si construiește un document agnostic de site.
3. Mesajele din coada agnostica sunt indexate de alta masina.
4. Unificatorul este si producator si consumator (pipeline).

Sisteme de cozi

1. Un tabel intr-o baza de date MySQL
2. Kestrel
3. Amazon SQS
4. etc

Cozi: Amazon SQS

- ❑ Implementare proprie (closed source)
- ❑ O coada este distribuite pe mai multe masini
- ❑ Gestionata de amazon
- ❑ Dimensiune maxima pe mesaj limitata (kb)
- ❑ Dupa citire un mesaj este marcat (nu sters) si nu este servit altui consumator pentru o perioada de timp. Mesajul trebuie sters de cel care il consuma!
- ❑ **!** Nu se garanteaza ca un mesaj este citit o singura data

Cozi: Amazon SQS

Create New Queue

Cancel

Please enter a name for your new queue. Queue names must be 1-80 characters in length and be composed of alphanumeric characters, hyphens (-), and underscores (_). Your queue will be created in the US West (Oregon) region.

Region: US West (Oregon)

Queue Name:

Configure your new queue by setting queue attributes (optional).

Default Visibility Timeout:	<input type="text" value="30"/>	<input type="button" value="seconds"/>	Value must be between 0 seconds and 12 hours.
Message Retention Period:	<input type="text" value="4"/>	<input type="button" value="days"/>	Value must be between 1 minute and 14 days.
Maximum Message Size:	<input type="text" value="256"/>	KB	Value must be between 1 and 256 KB.
Delivery Delay:	<input type="text" value="0"/>	<input type="button" value="seconds"/>	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time:	<input type="text" value="0"/>	seconds	Value must be between 0 and 20 seconds.

Cancel

Create Queue

Cozi: Amazon SQS



The screenshot shows a dialog box titled "Send a Message to Coada-de-test" with a "Cancel" button and a close icon (X) in the top right corner. The main instruction reads "Enter the text of a message you want to send." Below this is a large text input area containing the text "mesaj". At the bottom, there is a checkbox labeled "Delay delivery of this message by" followed by a numeric input field with the value "0", a dropdown menu currently set to "seconds", and the text "(up to 15 minutes)". At the very bottom right, there are two buttons: "Cancel" and "Send Message".

Send a Message to Coada-de-test Cancel X

Enter the text of a message you want to send.




mesaj

☐ Delay delivery of this message by 0 seconds (up to 15 minutes).

Cancel Send Message

Message Retention Period: 4 days

Cozi: Amazon SQS

Queues				
 Create New Queue		Queue Actions ▼		 Show/Hide  Refresh
Filter by Prefix: <input type="text"/>			1 to 1 of 1 items	
	Name	Messages Available	Messages in Flight	Created
<input checked="" type="checkbox"/>	Coad-a-de-test	3	0	2013-11-07 22:18:36 GMT+02:00

Auto scalare



- Un sistem care auto gestioneaza puterea sa de calcul
- Monitorizeaza propriile statistici si reactioneaza la schimbari adaugand sau stergand resurse (masini)

Auto scalare: alarme

- Media de utilizare a procesorului in grupul de lucratori este mai mare de 70%.
- Numarul de mesaje din coada este mai mare de 50 timp de o ora.
- Numarul de mesaje din coada este mai mic de 50 timp de o ora.

Auto scalare: AWS

- Amazon ofera un serviciu de autoscalare, dar nu are interfata web. Se poate accesa folosind clientul din linia de comanda oferit de amazon.
- `aws create-launch-configuration` creaza o configuratie de start pentru un grup. Contine informatii legate de tipul masinii si ce OS (AMI) va rula ea.
- `aws create-auto-scaling-group` creaza o configuratie de autoscalare. Contine informatii despre launch-configuration precum si valorile minime si maxime pentru numarul de masini.

Auto scalare AWS: Documentatia CLI

- <http://docs.aws.amazon.com/cli/latest/reference/>

Java API

- ❑ AWS ofera o librerie (java, python ...) pentru a comunica cu serviciile din cloud.
- ❑ Toate api-urile folosit de consola web pot fi accesate folosind aceasta librerie.
- ❑ Contine si API-uri pentru componente care nu au interfata web.

Java API: Autentificare

- API-ul are nevoie de 2 chei pentru a autentifica si autoriza un utilizator: o cheie de access si una secreta.
- Libraria are suport pentru autentificare:

```
String accessKey = "abcdefgh...";  
String secretKey = "secret-hbdf6tsf6";
```

```
AWSCredentials cred = new BasicAWSCredentials(  
    accessKey, secretKey  
)
```

Java API: Trimiterea unui mesaj in SQS

```
AmazonSQSClient sqs = new AmazonSQSClient(cred);  
SendMessageRequest request = new  
    SendMessageRequest("adresa-coada", "mesaj1");  
SendMessageResult result =  
    sqs.sendMessage(request);
```

Java API: Documentatie

- <http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/>