

unicorn



AWS Meetup Québec

Fonctionnement et remerciements



unicorn



AWS SQS / Lambda

Lambda Concurrency Limits and SQS Triggers



Jean-Philippe Lemelin

Chief Technology Officer, Co-Founder @ Unicorn





Introduction

- Chez Unicorn, on est des fans du serverless

22 / 02 / 2023



DynamoDB



Lambda



S3 Bucket



SES



CloudWatch



SQS



EventBridge



Cognito



API Gateway

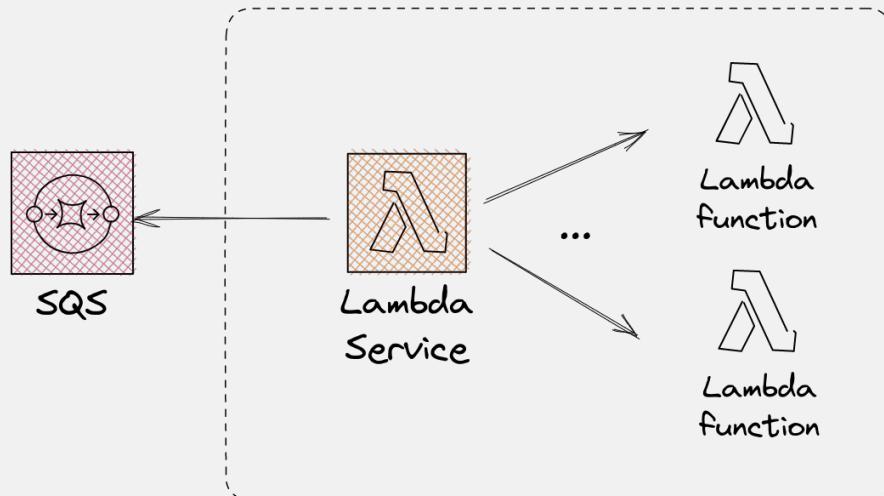


CloudFront



SQS Lambda

22 / 02 / 2023



Lambda service scales the number of invocations based on the number of messages



SQS Scaling



22 / 02 / 2023

“ Lambda reads up to five batches and sends them to your function. If messages are still available, Lambda increases the number of processes that are reading batches by up to 60 more instances per minute. The maximum number of batches that can be processed simultaneously by an event source mapping is 1000. ”



Parfait 🥂

- La limite est à 1000 exécutions concurrentes

Concurrent executions

Details			
Description			
Quota code	Quota ARN		
L-B99A9384	arn:aws:servicequotas:ca-central-1:392199159898:lambda/L-B99A9384		
Utilization	Applied quota value	AWS default quota value	Adjustable
0	1,000	1,000	Yes



Comment limiter le nombre de Lambdas concurrentes?

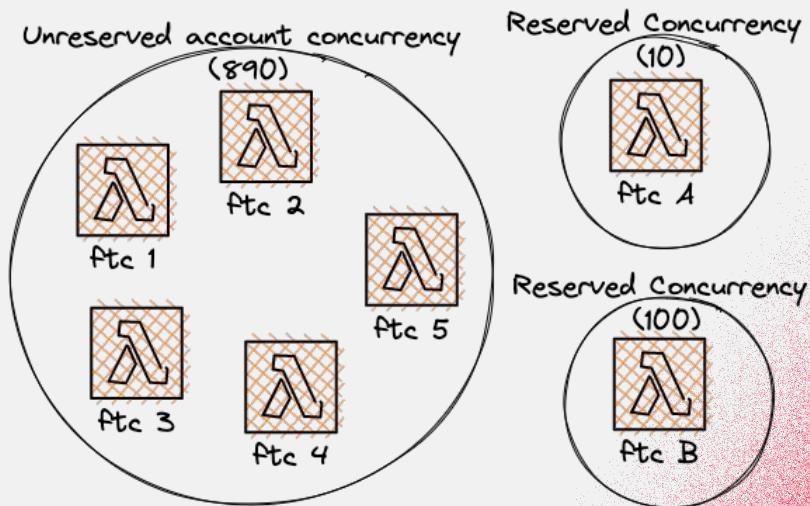


Lambda Reserved Concurrency



Reserved Concurrency

- Par défaut, AWS Lambda nous donne un pool de 1000 exécutions concurrent par account
- Toutes les lambda utilisent ce pool
- **Reserved Concurrency** permet de définir un autre pool alternatif pour une lambda
- Réduit la capacité de toutes les autres lambda





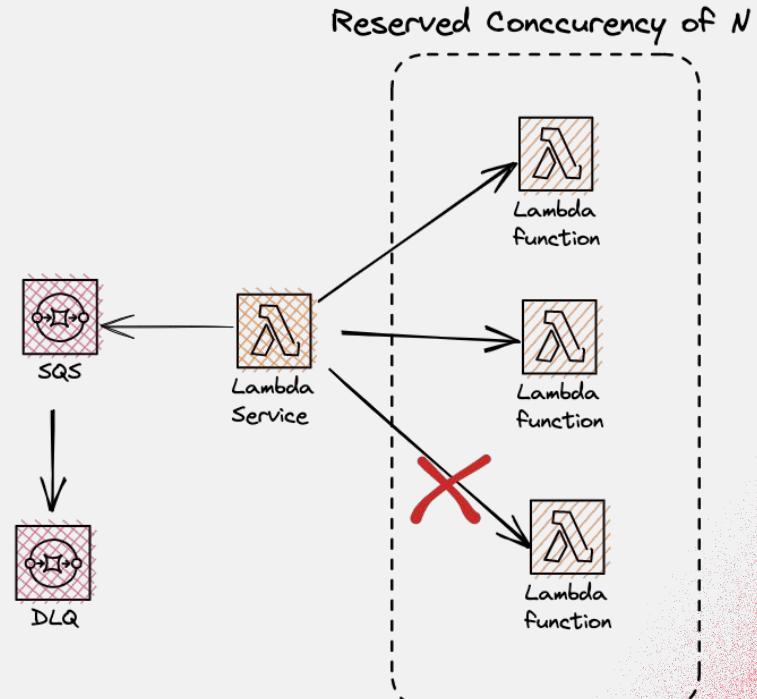
Interaction avec SQS

22 / 02 / 2023

Aucun impact sur le pattern "SQS => Lambda"

Le scaling des **pooler** va quand même avoir lieu du côté du Lambda Service 🤖

Problème: des messages se retrouvent dans la DLQ pour cause de **throttling** et non d'un problème d'exécution



The $N+1$ message is returned to the SQS queue



Piste de solution

Selon la doc de AWS

- visibilityTimeout = 6x le timeout de la fonction
 - maxReceiveCount = 5
- Avoir une DLQ qui retient les messages assez longtemps pour nous permettre de les remettre dans la queue

Configuring a queue to use with Lambda

Create an SQS queue to serve as an event source for your Lambda function. Then configure the queue to allow time for your Lambda function to process each batch of events—and for Lambda to retry in response to throttling errors as it scales up.

To allow your function time to process each batch of records, set the source queue's [visibility timeout](#) to at least six times the [timeout that you configure](#) on your function. The extra time allows for Lambda to retry if your function is throttled while processing a previous batch.

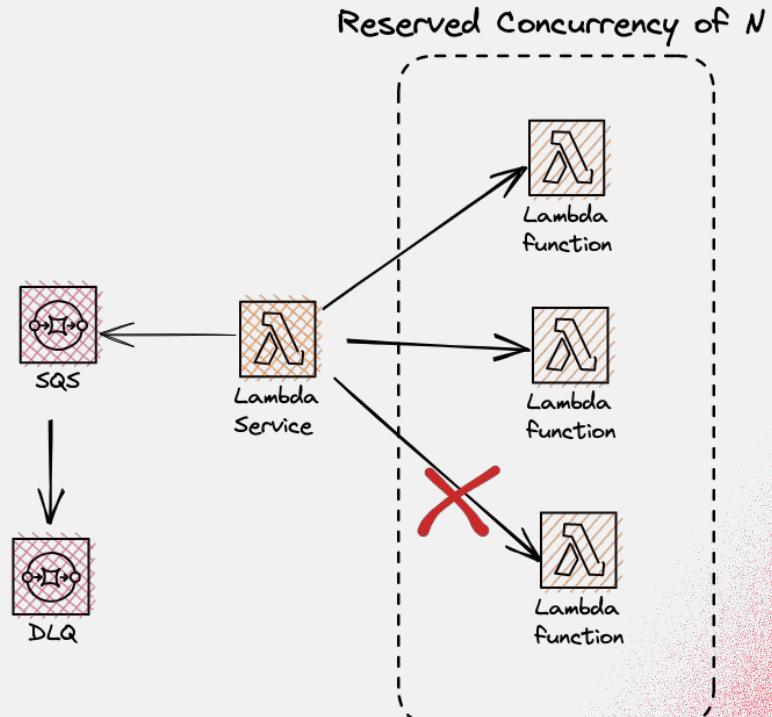
If your function fails to process a message multiple times, Amazon SQS can send it to a [dead-letter queue](#). If your function returns an error, all items in the batch return to the queue. After the [visibility timeout](#) occurs, Lambda receives the message again. To send messages to a second queue after a number of receives, configure a dead-letter queue on your source queue.

If your function returns an error, or can't be invoked because it's at maximum concurrency, processing might succeed with additional attempts. To give messages a better chance to be processed before sending them to the dead-letter queue, set the [maxReceiveCount](#) on the source queue's [redrive policy](#) to at least 5.



Limitations

- Des **bons** messages peuvent quand même arriver dans la DLQ
-



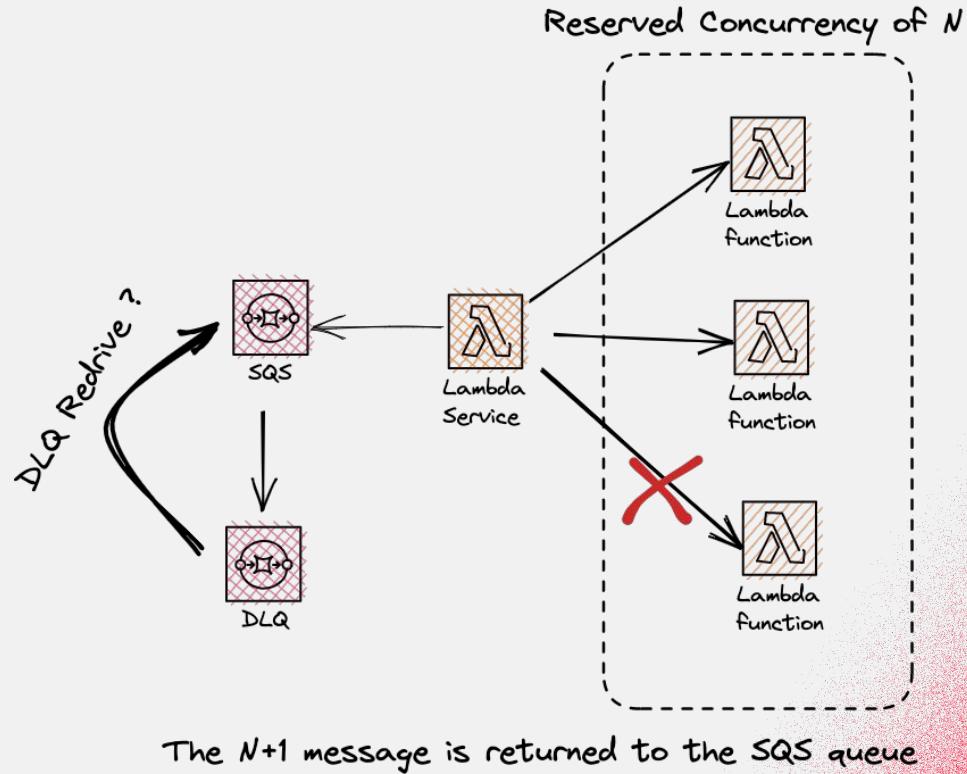
The $N+1$ message is returned to the SQS queue



Piste de solution

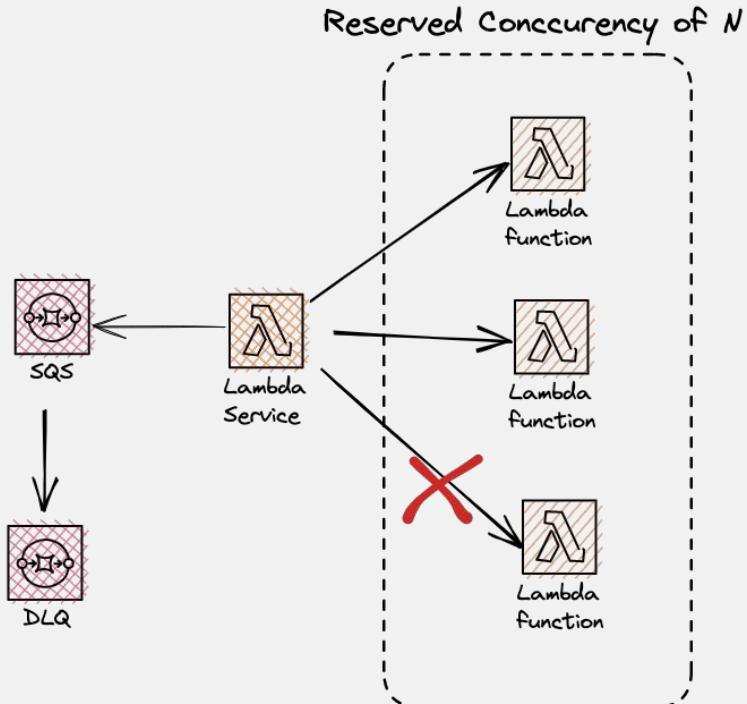
Mécanisme pour différencier les messages qui se sont rendus ici à cause du **throttling**

Ça ajoute de la complexité!





DEMO



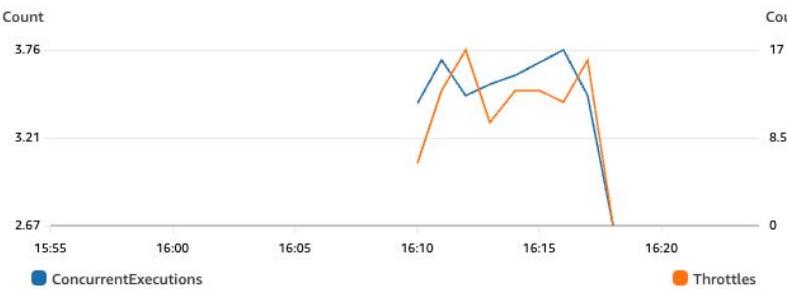
The $N+1$ message is returned to the SQS queue



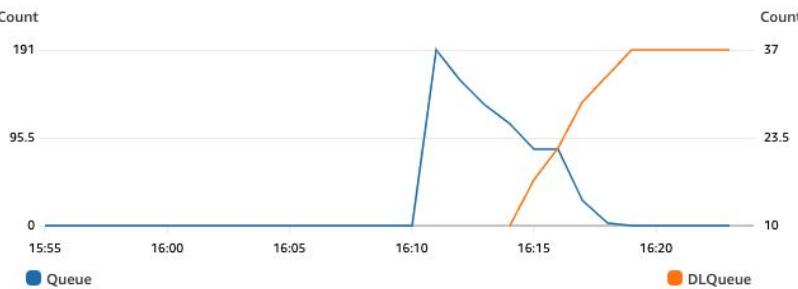
22 / 02 / 2023

Reserved Concurrency

Lambda Function



Number of msg SQS vs DLQ





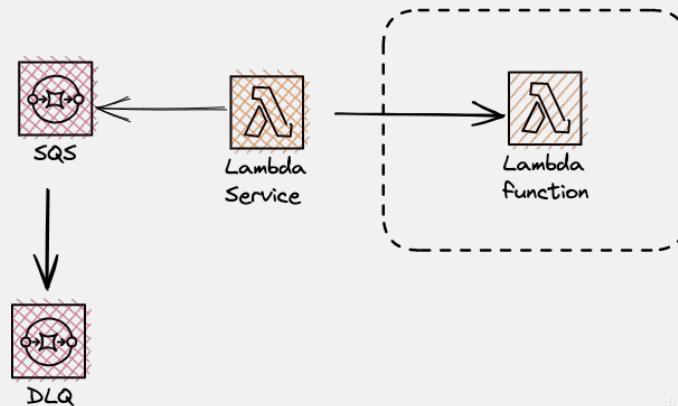
SQS FIFO



Piste de solution

Utiliser une SQS de type FIFO

- Permet d'avoir 1 seul lambda
- Que faire si on en veut 2 ou plus?



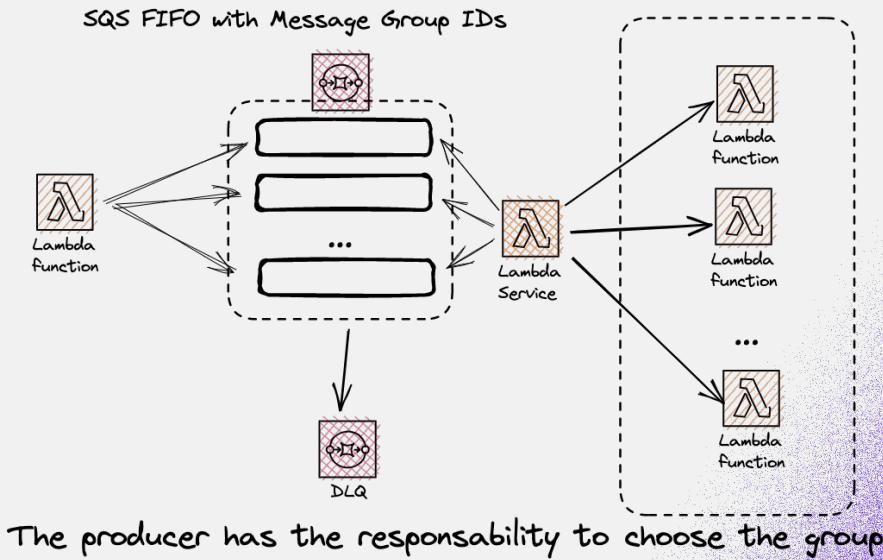


Piste de solution

22 / 02 / 2023

Utiliser les SQS FIFO avec les **Group IDs**

- 1 **pooler** par groupe
- C'est comme avec des mini queues avec 1 seul worker
- Le choix du group se fait lors de l'envoi du message dans la queue par le **producteur**

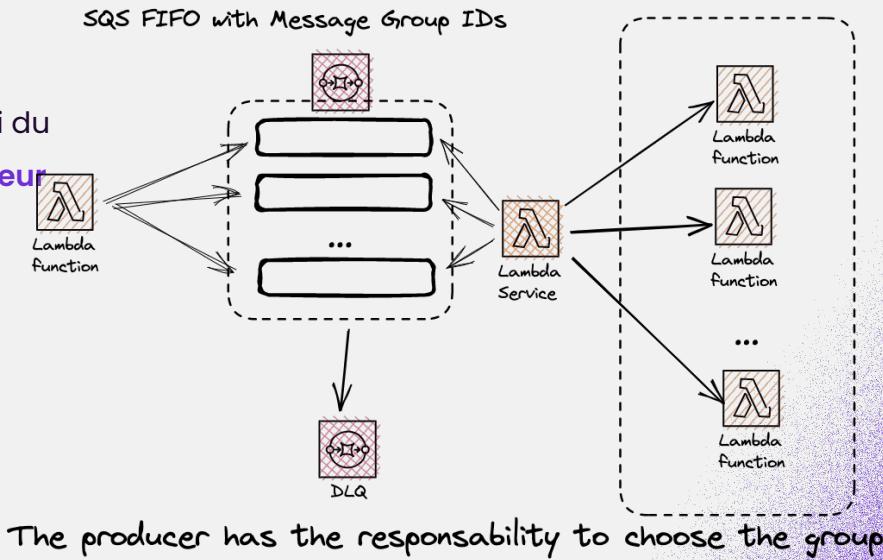




Limitations

22 / 02 / 2023

- Le choix du groupe se fait lors de l'envoi du message dans la queue par le **producteur**
- Aucun rebalancement automatique
- Fonction pour déterminer le Group ID
 - random % N
- 300-3000 Operations /sec
- Pas de **DLQ Redrive**

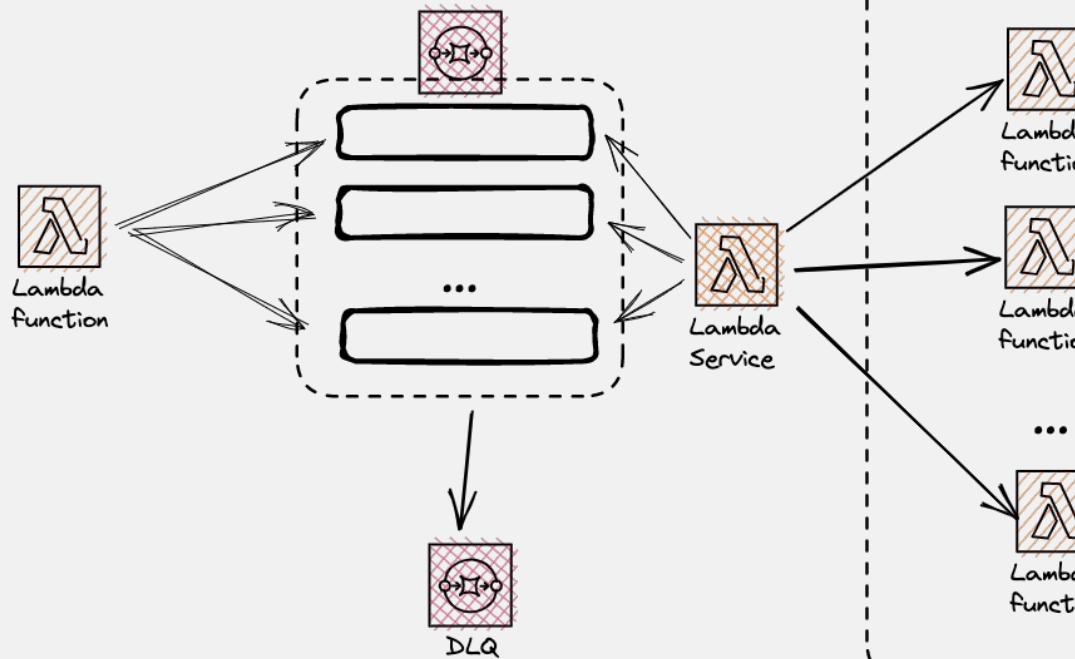




DEMO



SQS FIFO with Message Group IDs

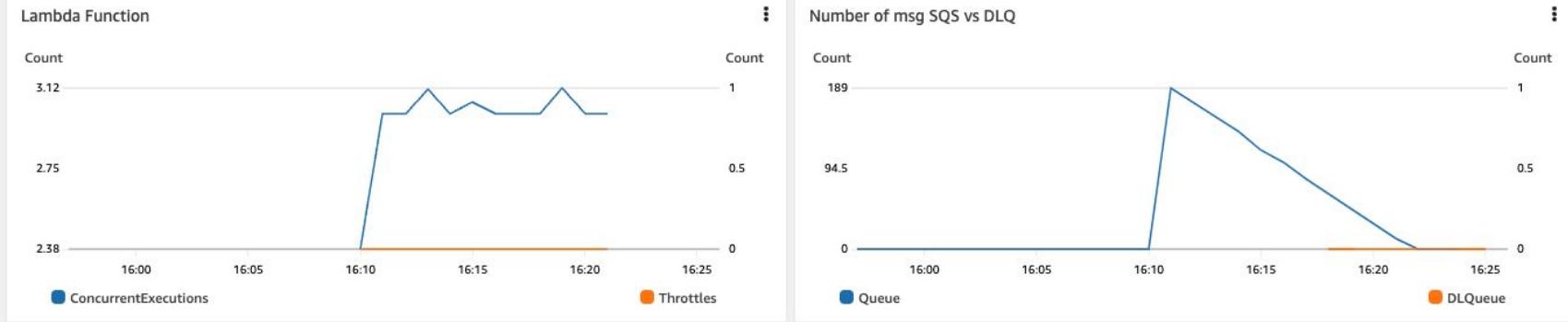


The producer has the responsibility to choose the group



22 / 02 / 2023

SQS Fifo





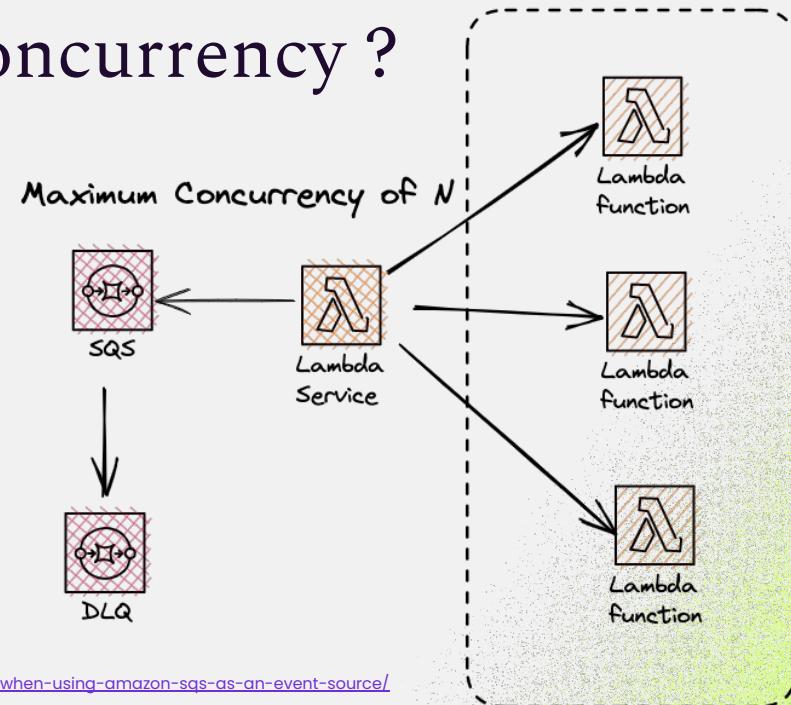
Maximum Lambda Concurrency



Maximum Lambda Concurrency ?

22 / 02 / 2023

- Annoncé le 12 Janvier 2023
- Lambda service, les **poolers** vont respecter le **Maximum Concurrency**

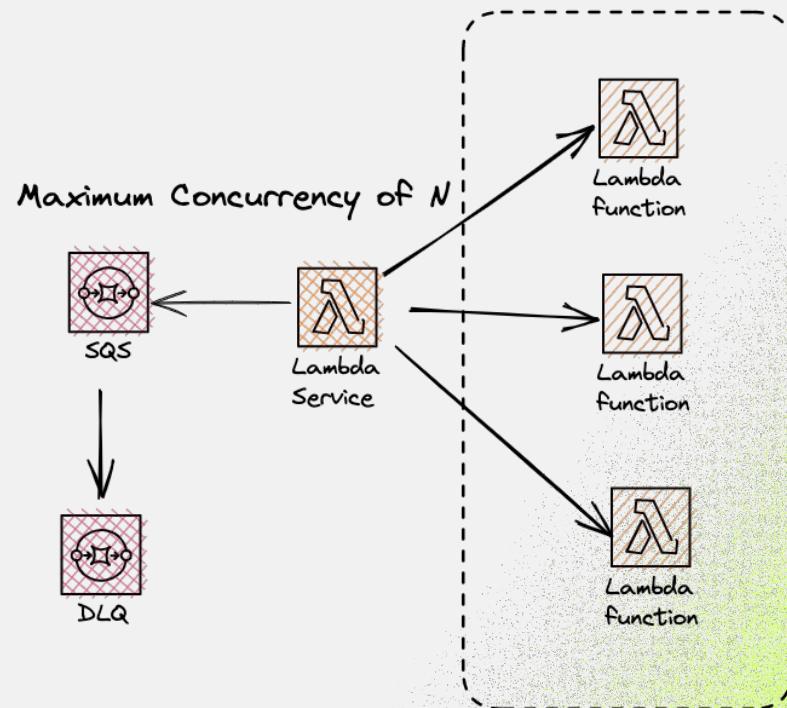




Limitation

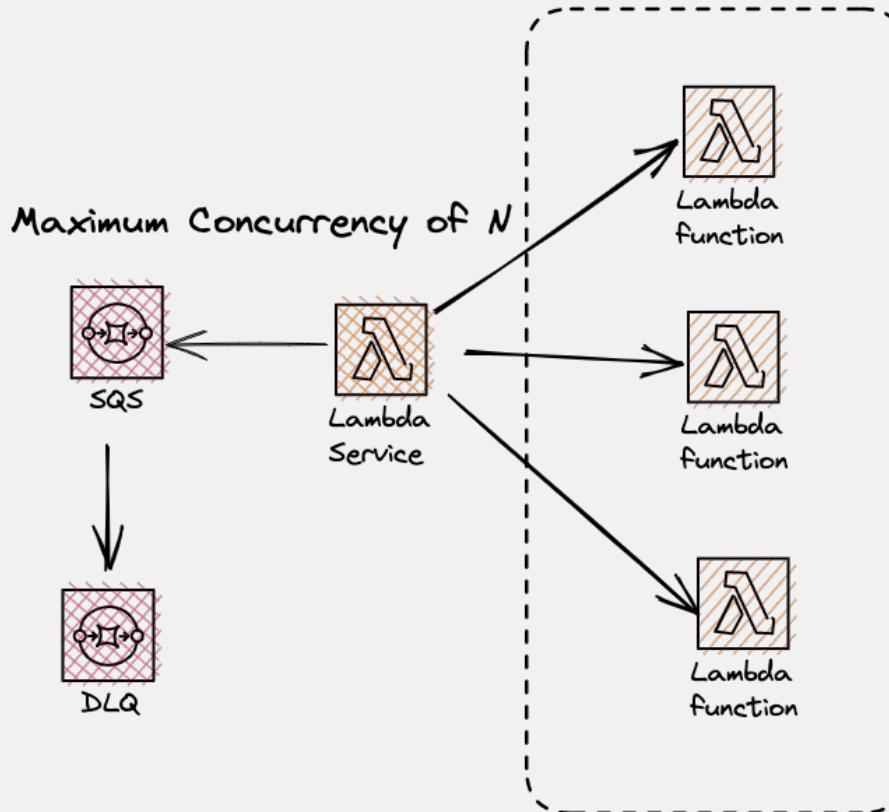
- Le minimum est **2**, pas 1
- ✓ SQS FIFO dans ce cas 😊

22 / 02 / 2023



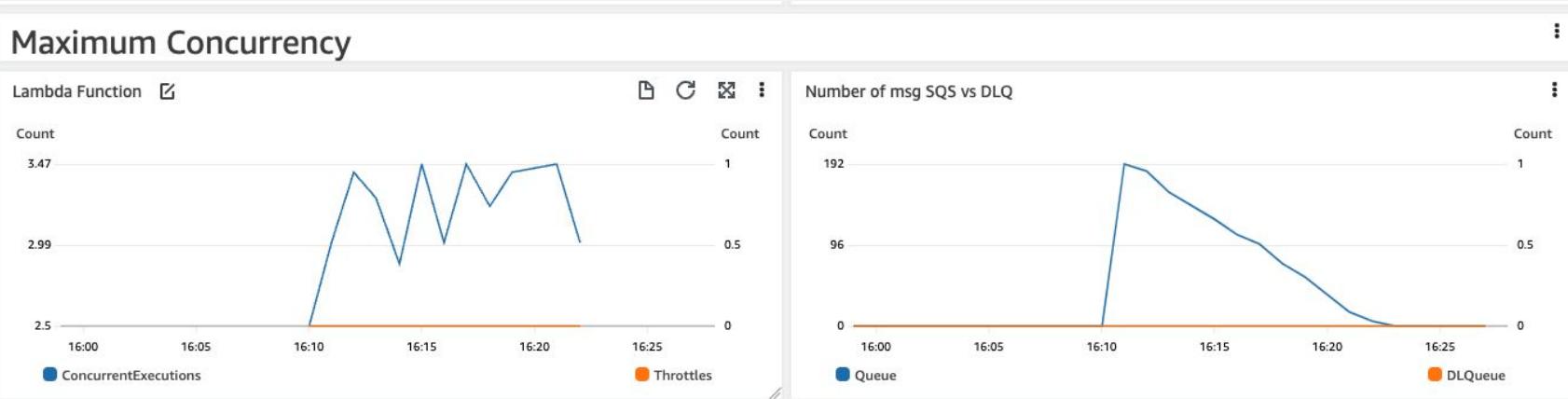


DEMO





22 / 02 / 2023





Trop beau pour être vrai?



Non, mais...

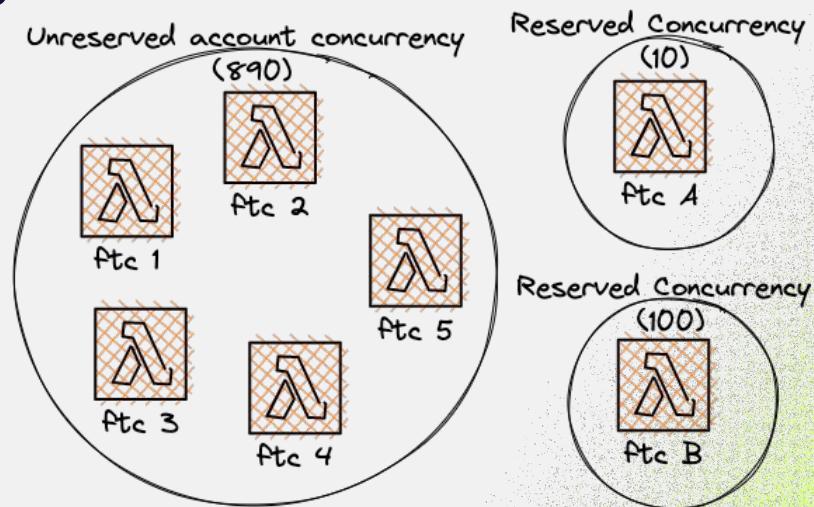
- Si le **Maximum Concurrency** est plus grand que le **Available Concurrency**



Available Concurrency

22 / 02 / 2023

- **Reserved concurrency**
 - Le Maximum Concurrency doit être idéalement égal
- **Unreserved concurrency**
 - Ça dépend 😊





Details

Description

The maximum number of events that functions can process simultaneously in the current Region.

Quota code

L-B99A9384

Quota ARN

arn:aws:servicequotas:ca-central-1:660166667835:lambda/L-B99A9384

Utilization

0

Applied quota value

250

AWS default quota value

1,000

Adjustable

Yes



Conclusion



Conclusion

En 2023, le **maximum concurrency** est vraiment la solution que la communauté attendait depuis des années!



Next Step

- Mettre à jour vos infra pour utiliser le **Maximum Concurrency**
- Mettre à jour la documentation AWS?



Questions?

<https://github.com/UnicorneCloud/AwsMeetupSQS>

