

# LMAX – Architecture

Thomas LINCK

# Sommaire

Exigence et contexte du marché des changes

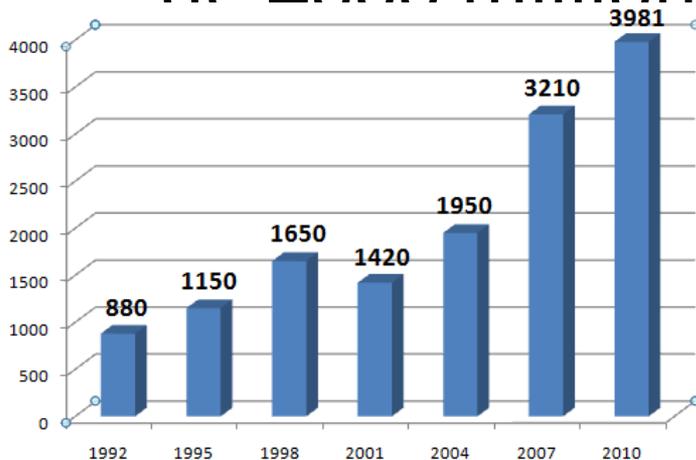
LMAX, origine du besoin

Quelques conseils selon LMAX

La solution : le pattern Disruptor

# FOReign EXchange

- Marché des changes mondial
- Taux de change variable mais pas si volatile que ça...
- Volumes importants, près de 4000 milliards de dollars



| Marché             | Volatilité |
|--------------------|------------|
| EUR/USD            | 1,31%      |
| FOREX (42 parités) | 1,46%      |
| T-BOND             | 1,38%      |
| CAC40              | 2,96%      |
| DOW JONES          | 2,70%      |
| NASDAQ             | 2,98%      |
| OR                 | 3,06%      |
| ACTION DANONE      | 3,60%      |
| ACTION TOTAL       | 3,64%      |
| BLE                | 4,77%      |
| PETROLE            | 4,90%      |

# LMAX

- London Multi Asset eXchange
  - Filiale de BetFair
  - Spécialisé dans le passage d'ordres de bourse
  - Exécution des ordres rapide et anonyme
  - Agrée par la Financial Services Authority

# Besoins d'une plateforme de trading

- Latence faible



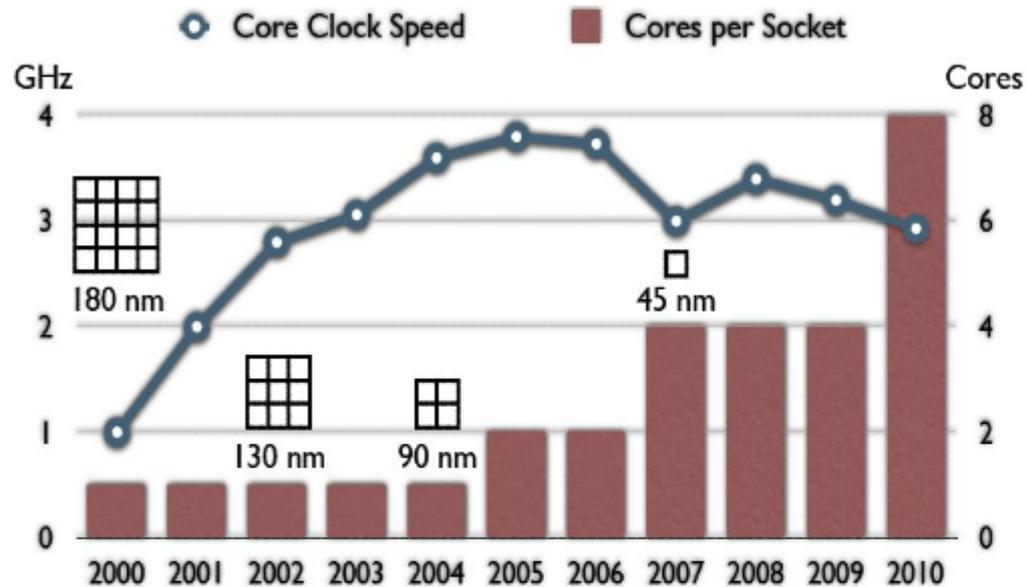
- Dépendance faible

# Quelques Conseils

- Utiliser autant que possible la « Mechanical Sympathy »
- Garder le jeu de travail en mémoire
- Ecrire du code propre et compact
- Prendre le temps nécessaire à la modélisation du domaine
- Utiliser la concurrence pour les bonnes raisons

# Mechanical Sympathy

- La guerre des GHz a pris fin vers 2005.
- La mémoire a augmenté mais pas la latence



# Mechanical Sympathy

- Réseaux
  - 20 - 100 Mbits / secondes
  - Moins de 10 microsecondes pour un saut local
- Stockage
  - 1 To pour ~ 50€ (avant les inondations en Thaïlande)
  - Disques SSD

# Mechanical Sympathy

- Jackie Stewart – Pilote 3 fois champion du monde de F1
- *"The most amazing achievement of the computer software industry is its continuing cancellation of the steady and staggering gains made by the computer hardware industry."* Henry Peteroski

# Garder le jeu de travail en mémoire

- Garder les données et les comportements ensembles, les rapprocher (en terme de délais d'accès)
- Se permettre des interactions riches à des latences faibles

# Ecrire du code propre et compact

- « Any intelligent fool can make things bigger and more complex... It takes a touch of genius - and a lot of courage to move in the opposite direction.» Albert Einstein
- La complexité du code vient de la mauvaise compréhension du domaine métier

# Prendre le temps nécessaire à la modélisation du domaine

- Responsabilité unique : une classe -> une chose, une méthode -> une chose
- Connaissance des structures de données et la cardinalité des relations
- Les relations font très bien le travail

# Concurrence

- Deux principes doivent être respectés:
  - Exclusion mutuelle
  - Visibilité des modifications

# Et avec tous ces conseils??

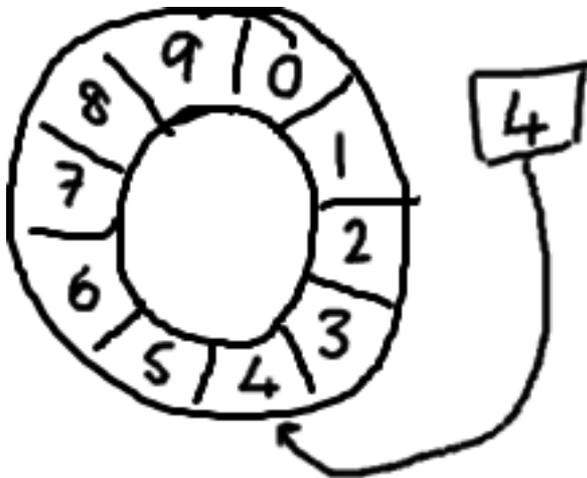
- 1 thread -> ~ 3 milliards d'instructions par seconde
- 10K TPS en ne faisant rien
- 100K TPS avec un code propre et bien organisé
- 1M TPS avec des tests de performance, et un domaine très bien modélisé.
- Ecrire un test de performance est plus compliqué qu'écrire le code cible lui-même.

# Disruptor

- Problèmes liés à la trop forte corrélation entre producteur, consommateurs et stockage des données générant des latences trop élevées.
- Alternative haute performance aux files d'attente bornées pour l'échange de données entre threads concurrents.

# Disruptor – Ring Buffer

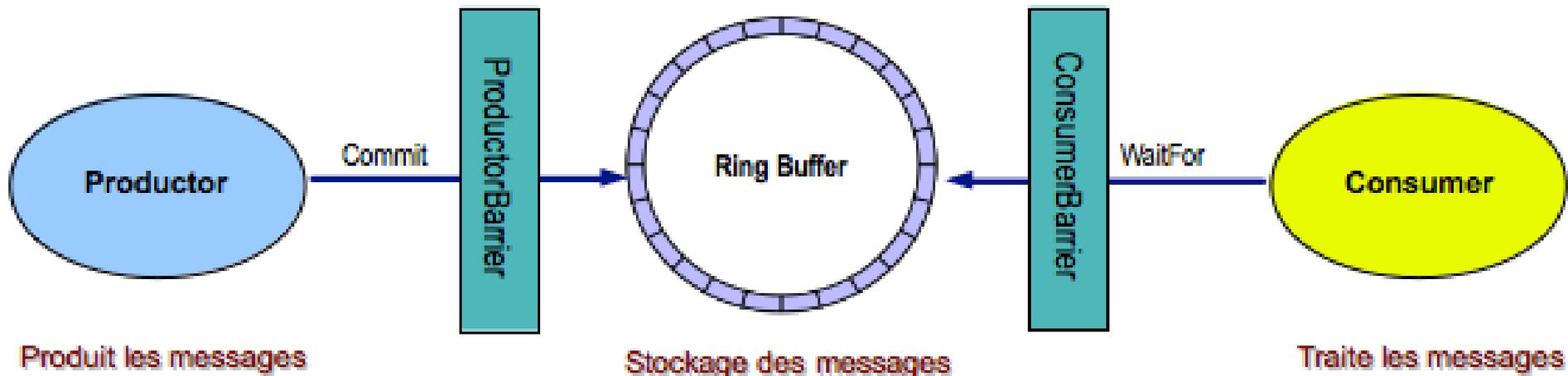
- Utilisé pour passer les données d'un contexte à un autre.
- Pré-allocation de la mémoire du ring buffer.



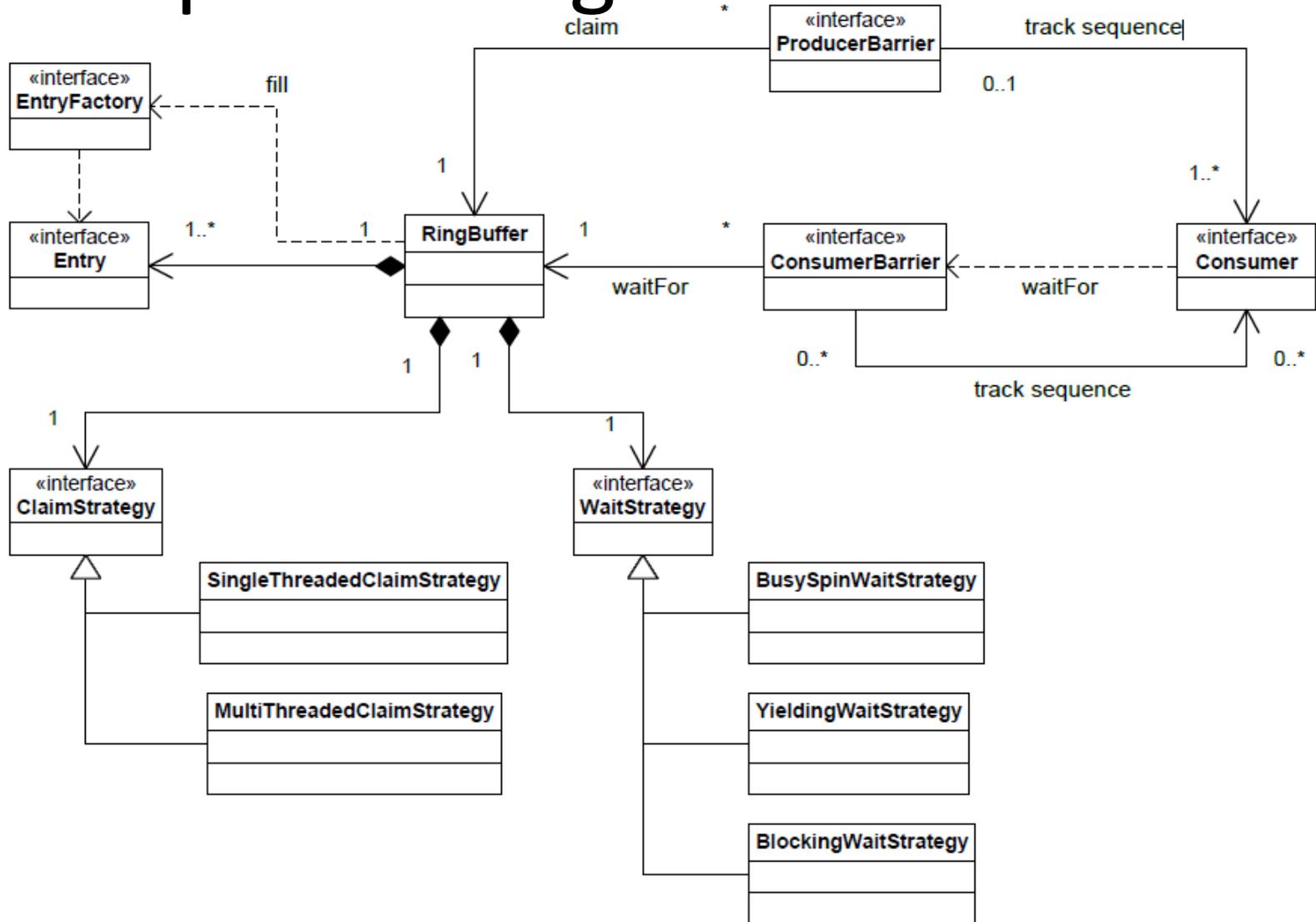
- *Pas de GC car les modifications sont faites en place.*
- *Modèle prévisible d'accès*

# Disruptor – Schéma simple

- Barrière de gestion instanciées par le RingBuffer.



# Disruptor – Diagramme de classes



# Disruptor - Tests

- Tests avec un processeur de 2.2Ghz  
i7-2720QM
- 50 millions d'injections de paquets à 1 micro  
seconde d'intervalle

|                       | ArrayBlockingQueue (ms) | Disruptor (ms) | Ratio  |
|-----------------------|-------------------------|----------------|--------|
| Latence minimale      | 145                     | 29             | 5      |
| Latence significative | 32 757                  | 52             | 630    |
| 99% sont sous         | 2 097 152               | 128            | 16 400 |
| 99.99% sont sous      | 4 194 304               | 8 192          | 512    |
| Latence maximale      | 5 069 086               | 175 567        | 29     |

# Avez-vous des questions?

- <http://www.lmaxtrader.co.uk/>
  - Site web commercial de LMAX
- <http://mechanical-sympathy.blogspot.com/>
  - Blog sur la « Mechanical Sympathy »
- <http://code.google.com/p/disruptor>
  - Code source du Disruptor
- <http://mechanitis.blogspot.com>
  - Blog d'un développeur, riche en informations