

Paris JUG

Spring Batch

Mardi 14 Mai 2013

Olivier Bazoud
Julien Jakubowski



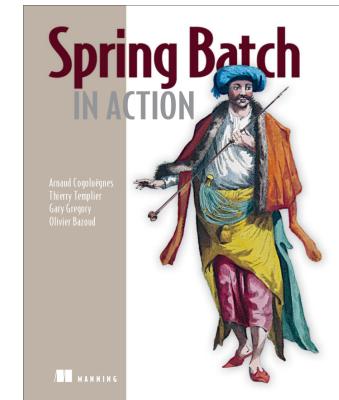
Intervenants

Olivier Bazoud @obazoud

Architecte technique senior

Java EE / Spring, Spring Batch, NoSQL, Node.js

Co-auteur de « Spring Batch in Action »



Julien Jakubowski @jak78

Java depuis > 10 ans

Spécialisé Java EE / Spring



Vos réactions: #springbatch #parisjug



De quoi va-t-on parler ce soir ?

- On va parler de batchs en Java



Spring Batch

- ... et aussi de bière



<http://www.flickr.com/photos/fromeyetopixel/2559391584/>



Nos objectifs



En sortant de la salle, vous savez:

- ✓ Si Spring Batch est utile pour vous (ou pas)
- ✓ Les notions principales de Spring Batch
 - ... et quelques notions avancées
- ✓ Faire un batch « Spring Batch » dès demain

<https://github.com/obazoud/spring-batch-sug>



Batch : de quoi parle-t-on ?

Batch processing = répétition de traitements sur un ensemble de données...



<http://www.flickr.com/photos/burnblue/308441464/>



Batch : de quoi parle-t-on ?

... potentiellement grands volumes...

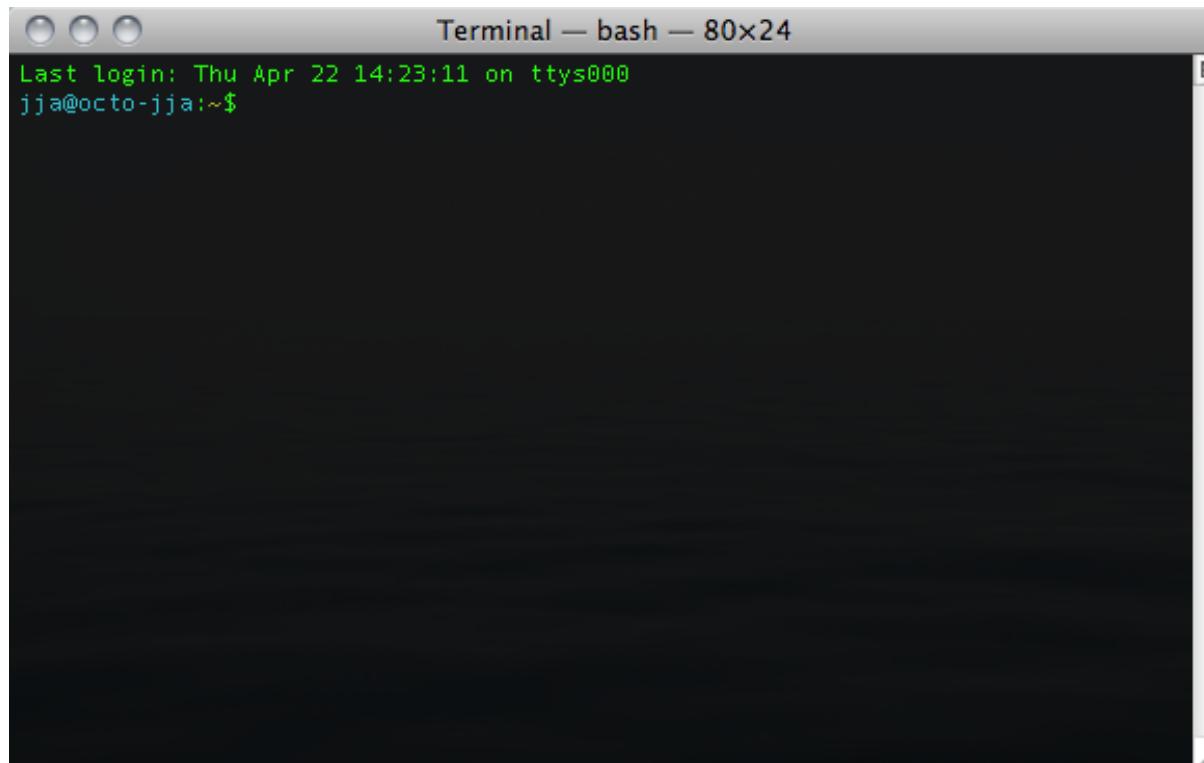


<http://www.flickr.com/photos/claudiasofia99/2878579560/>



Batch : de quoi parle-t-on ?

... sans intervention d'un utilisateur humain – pas d'interface homme-machine



Batch : de quoi parle-t-on ?

batch ≠ scheduler



Problèmes récurrents

- Fiabilité

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
  at org.jdom.ContentList.ensureCapacity(ContentList.java:355)
  at org.jdom.ContentList.add(ContentList.java:234)
  at org.jdom.ContentList.add(ContentList.java:131)
  at java.util.AbstractList.add(AbstractList.java:91)
  at org.jdom.Element.addContent(Element.java:811)
  at org.jdom.DefaultJDOMFactory.addContent(DefaultJDOMFactory.java:180)
  at org.jdom.input.SAXHandler.flushCharacters(SAXHandler.java:693)
  at org.jdom.input.SAXHandler.flushCharacters(SAXHandler.java:660)
  at org.jdom.input.SAXHandler.endElement(SAXHandler.java:716)
  at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.endElement(AbstractS
  at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanEndEle
  at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl$FragmentCo
  at com.sun.org.apache.xerces.internal.impl.XMLDocumentScannerImpl.next(XMLDocumentSc
  at com.sun.org.apache.xerces.internal.impl.XMLNSDocumentScannerImpl.next(XMLNSDocume
  at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanDocume
  at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configur
  at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configur
  at com.sun.org.apache.xerces.internal.parsers.XMLParser.parse(XMLParser.java:107)
  at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.parse(AbstractSAXPar
  at com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser.parse(SAXPars
  at org.jdom.input.SAXBuilder.build(SAXBuilder.java:489)
  at org.jdom.input.SAXBuilder.build(SAXBuilder.java:928)
  at fr.sug.springbatch.plainoldbatch.PlainOldBatch.run(PlainOldBatch.java:72)
  at fr.sug.springbatch.plainoldbatch.PlainOldBatch.main(PlainOldBatch.java:37)
```



Problèmes récurrents

- Maintenabilité



Problèmes récurrents

- Réinvention de la roue... carrée



Spring Batch propose...

- Un cadre
- Un vocabulaire (domain language)
- Traitement par lots
- Flow, Reprise sur erreur, ...
- Spring dans ses batchs
- ...

Parallélisme
Partitionnement

Spring Batch Admin



Spring Batch

Notions de base



BeerBatch

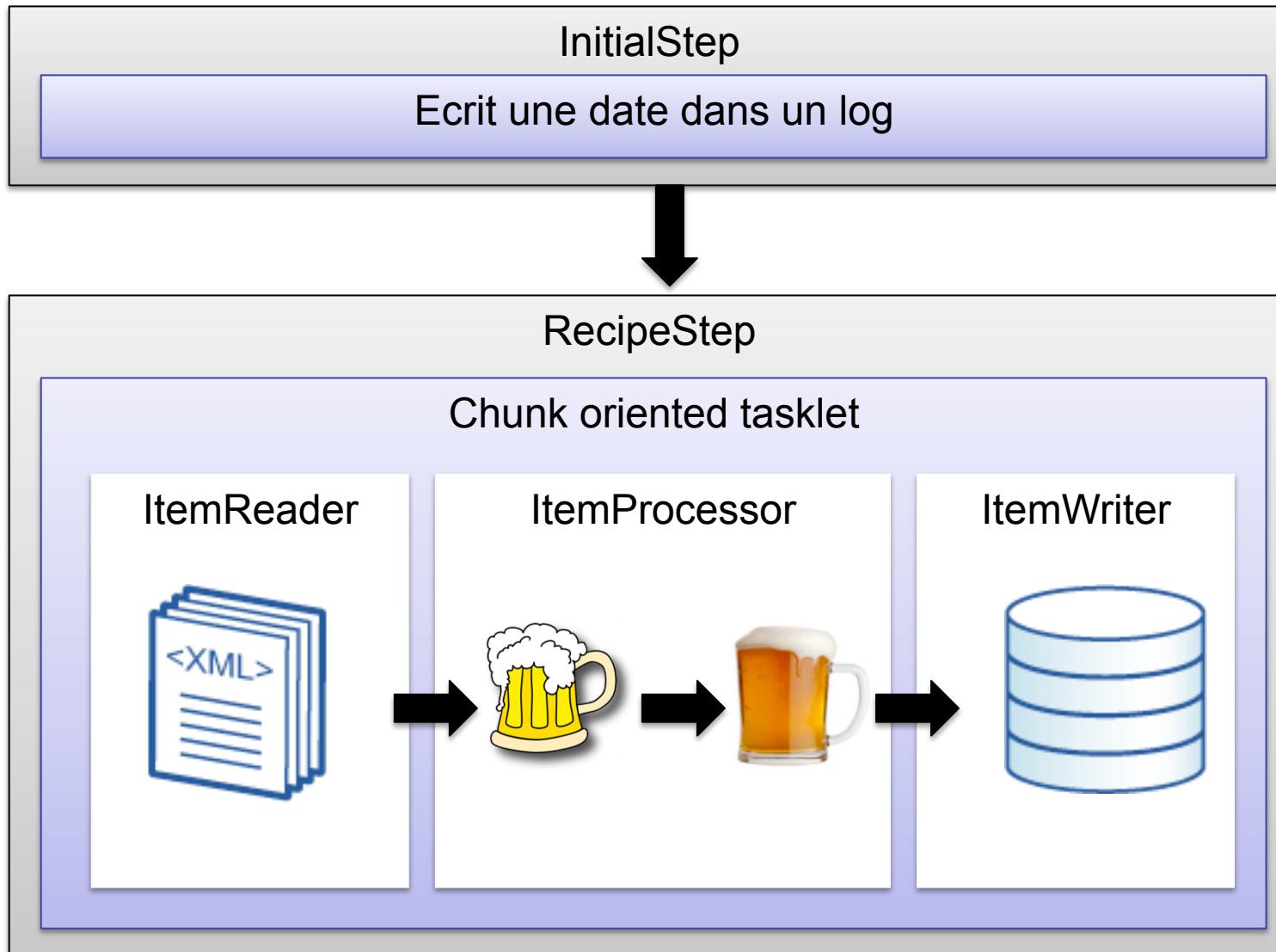
- Ecrire la date de début du batch
- Lire le fichier XML de recettes, au format BeerXML
- Ecrire en base de données



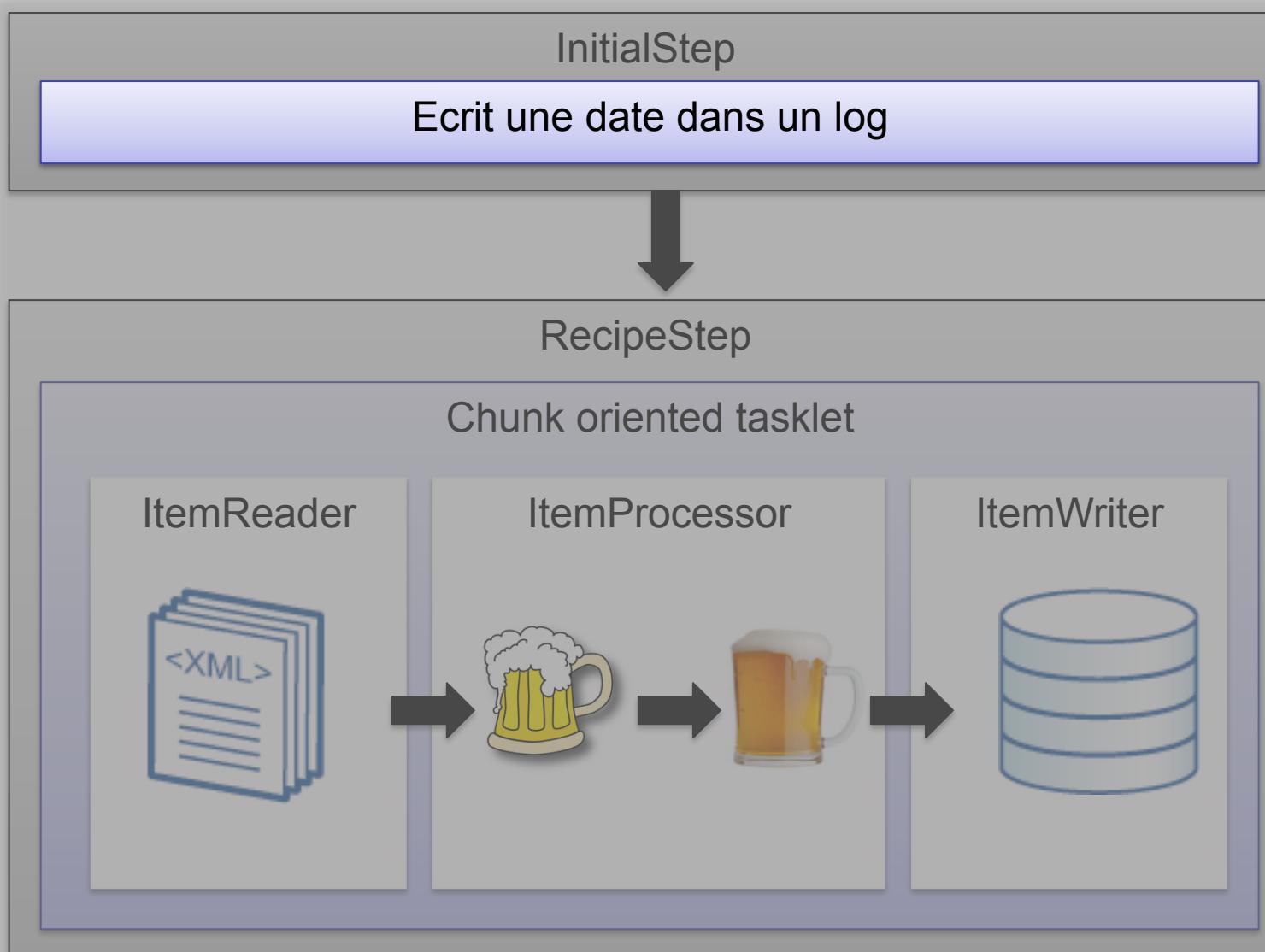
```
<RECIPES>
<RECIPE>
  <NAME>Burton Ale</NAME>
  <VERSION>1</VERSION>
  <TYPE>All Grain</TYPE>
  <BREWER>Brad Smith</BREWER>
  <ASST_BREWER></ASST_BREWER>
  <BATCH_SIZE>18.92716800</BATCH_SIZE>
  <BOIL_SIZE>20.81988500</BOIL_SIZE>
  <BOIL_TIME>60</BOIL_TIME>
  <EFFICIENCY>72.0</EFFICIENCY>
  <HOPS>
    <HOP>
      <NAME>Goldings, East Kent</NAME>
      <VERSION>1</VERSION>
      <ORIGIN>United Kingdom</ORIGIN>
      <ALPHA>5.50</ALPHA>
      <AMOUNT>0.0283500</AMOUNT>
      <USE>Boil</USE>
      <TIME>60.000</TIME>
      <NOTES>Used For: General purpose hops for bittering/finishing all
          British Ales
          Aroma: Floral, aromatic, earthy, slightly sweet spicy flavor
          Substitutes: Fuggles, BC Goldings
          Examples: Bass Pale Ale, Fullers ESB, Samuel Smith's Pale Ale
      </NOTES>
      <TYPE>Aroma</TYPE>
      <FORM>Pellet</FORM>
      <BETA>3.50</BETA>
      <HSI>35.0</HSI>
      <DISPLAY_AMOUNT>1.00 oz</DISPLAY_AMOUNT>
      <INVENTORY>1.00 oz</INVENTORY>
      <DISPLAY_TIME>60 min</DISPLAY_TIME>
    </HOP>
  </HOPS>
</RECIPE>
</RECIPES>
```



Schéma du batch



Tasklet



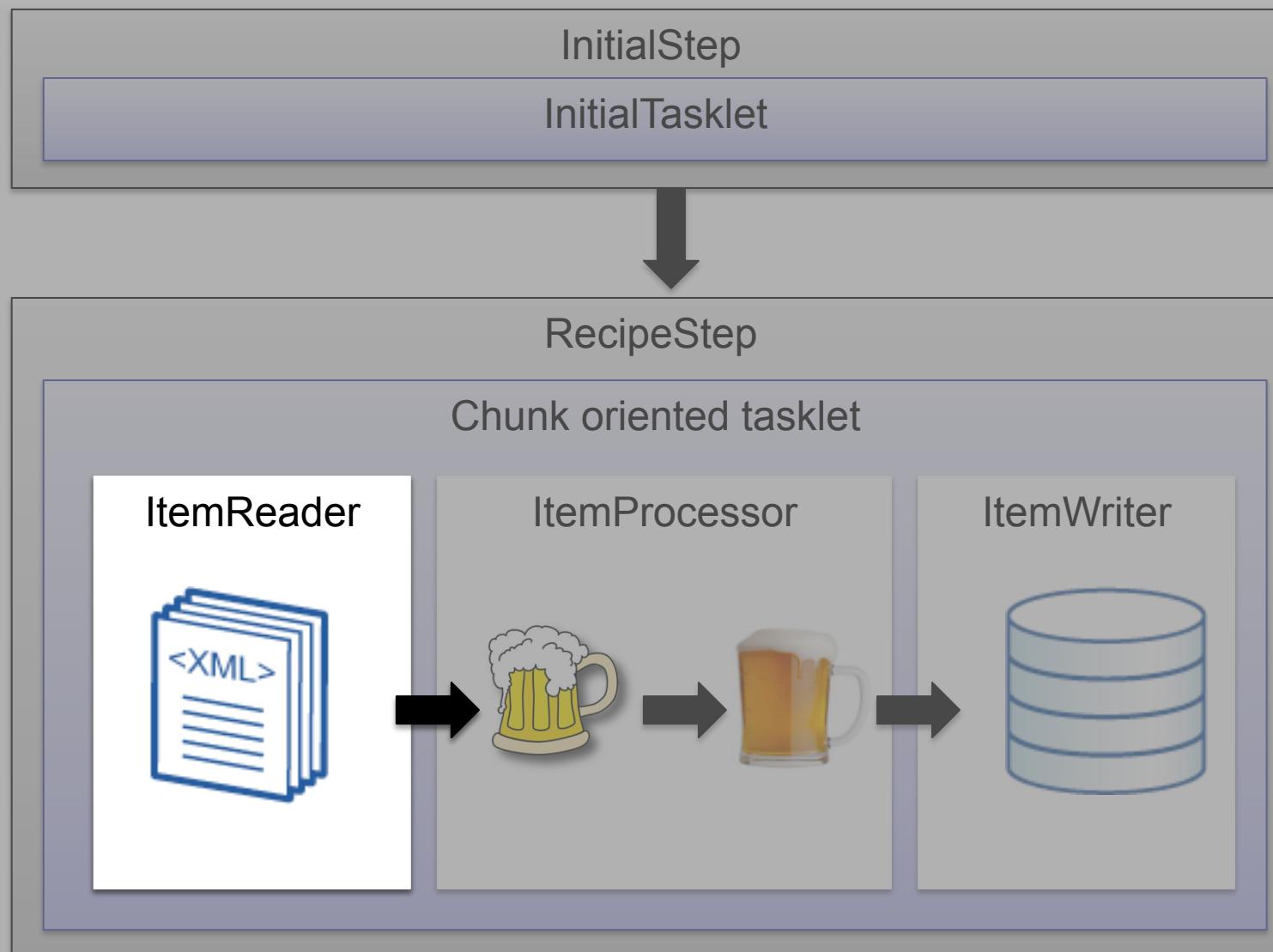
Tasklet

- Besoin
 - Effectuer 1 tâche unitaire
- Exemples
 - Unzip d'un fichier
 - Ecrire une date dans un log
 - ...

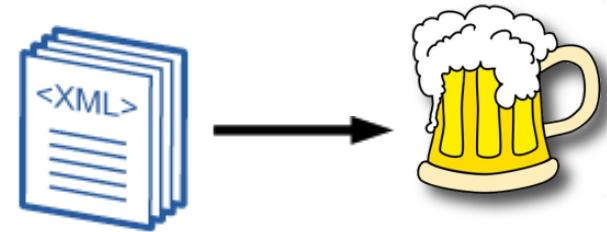
```
public interface Tasklet {  
  
    * Given the current context in the form of a step contribution, do whatever  
    RepeatStatus execute(StepContribution contribution, ChunkContext chunkContext) throws Exception;  
  
}
```



ItemReader



ItemReader

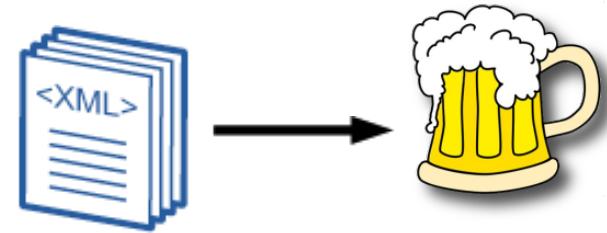


- Besoin
 - Lire le XML de recettes
- ItemReader
 - Fournit des items en entrée

```
public interface ItemReader<T> {  
  
    * Reads a piece of input data and advance to the next one. Implementations  
    T read() throws Exception, UnexpectedInputException, ParseException;  
  
}
```



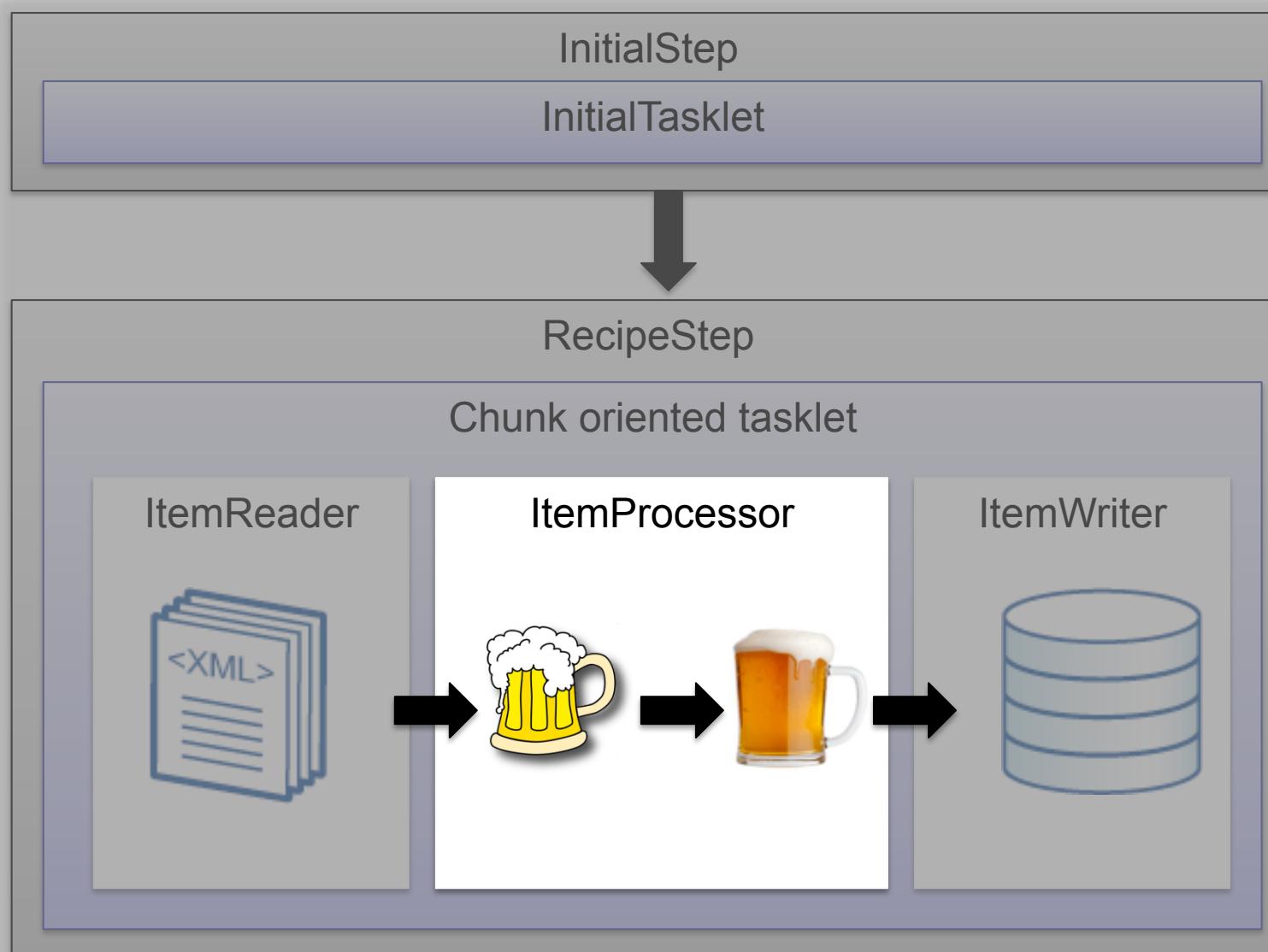
Morceaux de code



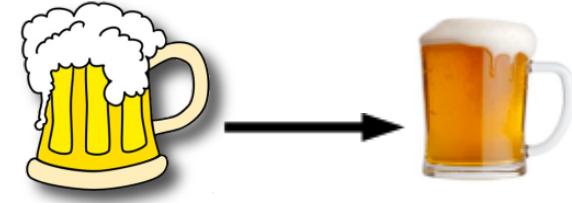
```
<bean id="recipesReader"
      class="org.springframework.batch.item.xml.StaxEventItemReader"
      scope="step">
    <property name="resource" value="#{jobParameters[recipes]}" />
    <property name="fragmentRootElementName" value="RECIPE" />
    <property name="unmarshaller" ref="recipeMarshaller" />
</bean>
```



ItemProcessor



ItemProcessor

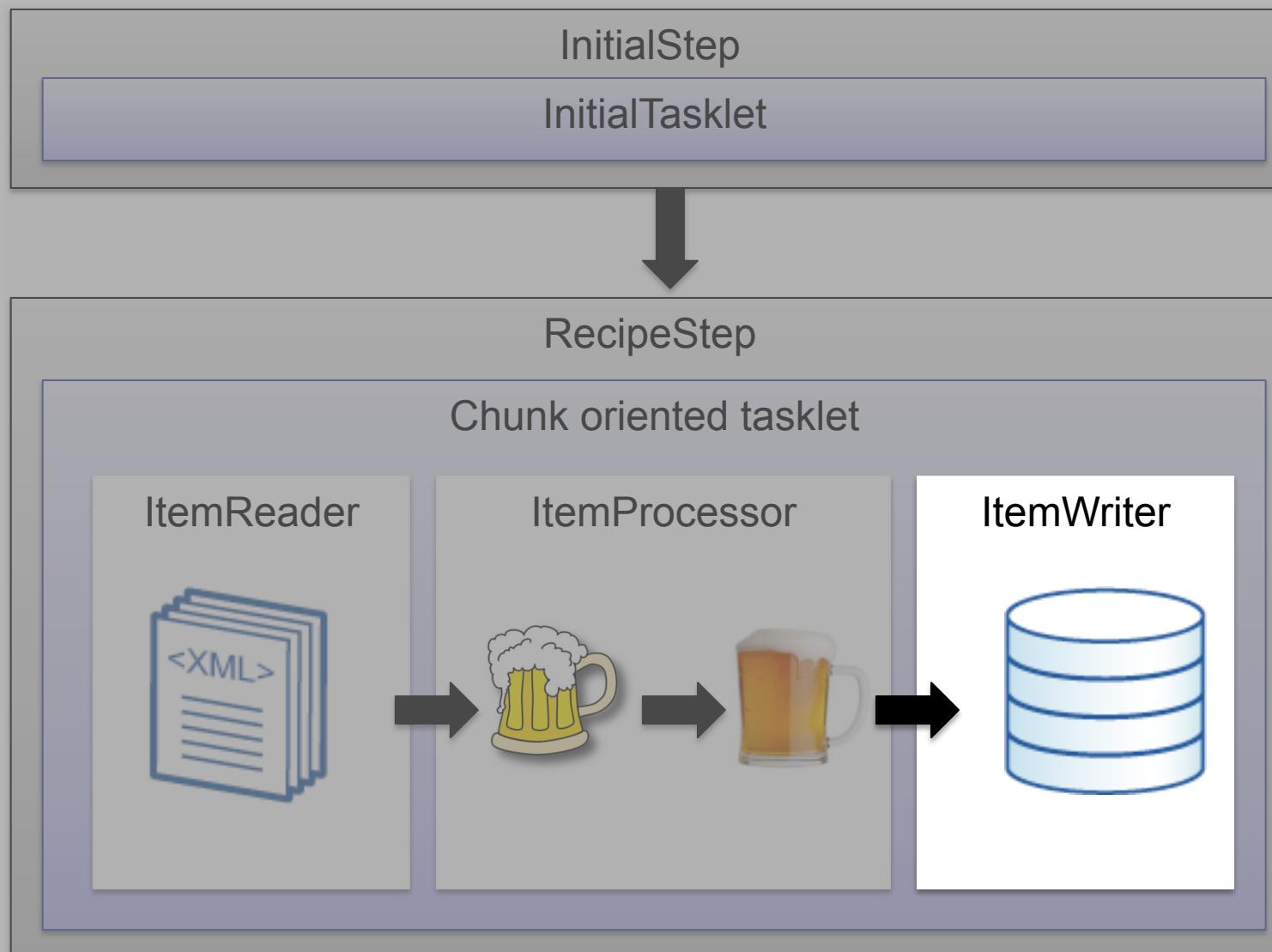


- Besoin
 - Transforme, valide et / ou filtre une recette
- ItemProcessor
 - Transforme un item en un autre
 - Filtrer ou rejeter un item
 - Emplacement pour les « règles métier »

```
public interface ItemProcessor<I, O> {  
  
    * Process the provided item, returning a potentially modified or new item for continued  
    O process(I item) throws Exception;  
}
```



ItemWriter



ItemWriter

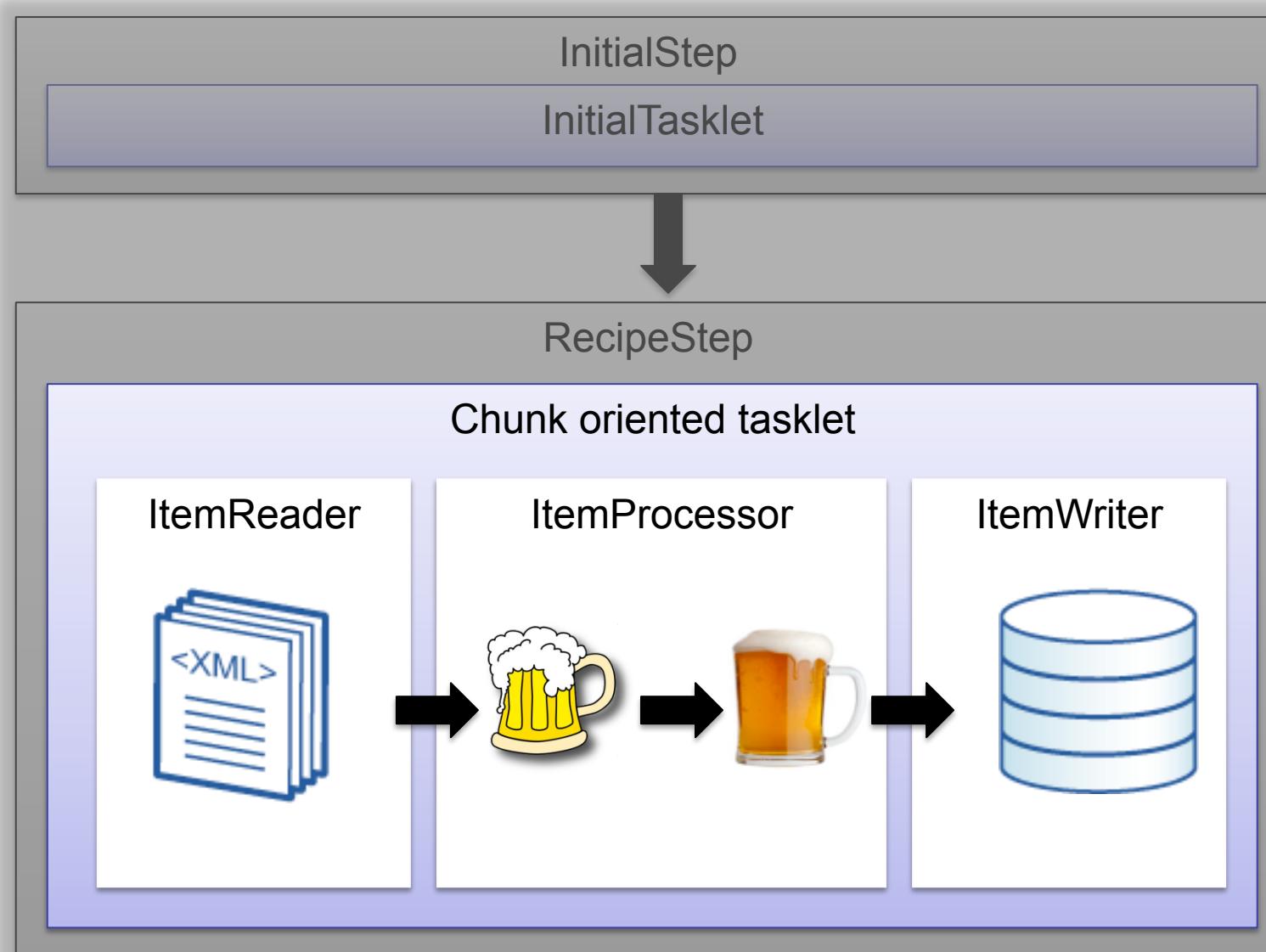


- Besoin
 - Décharge les bières dans une base SQL
- ItemWriter
 - Ecrit les items

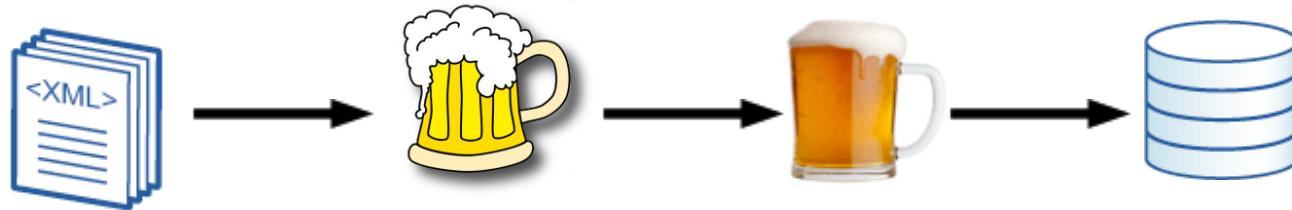
```
public interface ItemWriter<T> {  
  
    * Process the supplied data element. Will not be called with any null items  
    void write(List<? extends T> items) throws Exception;  
}
```



Chunk

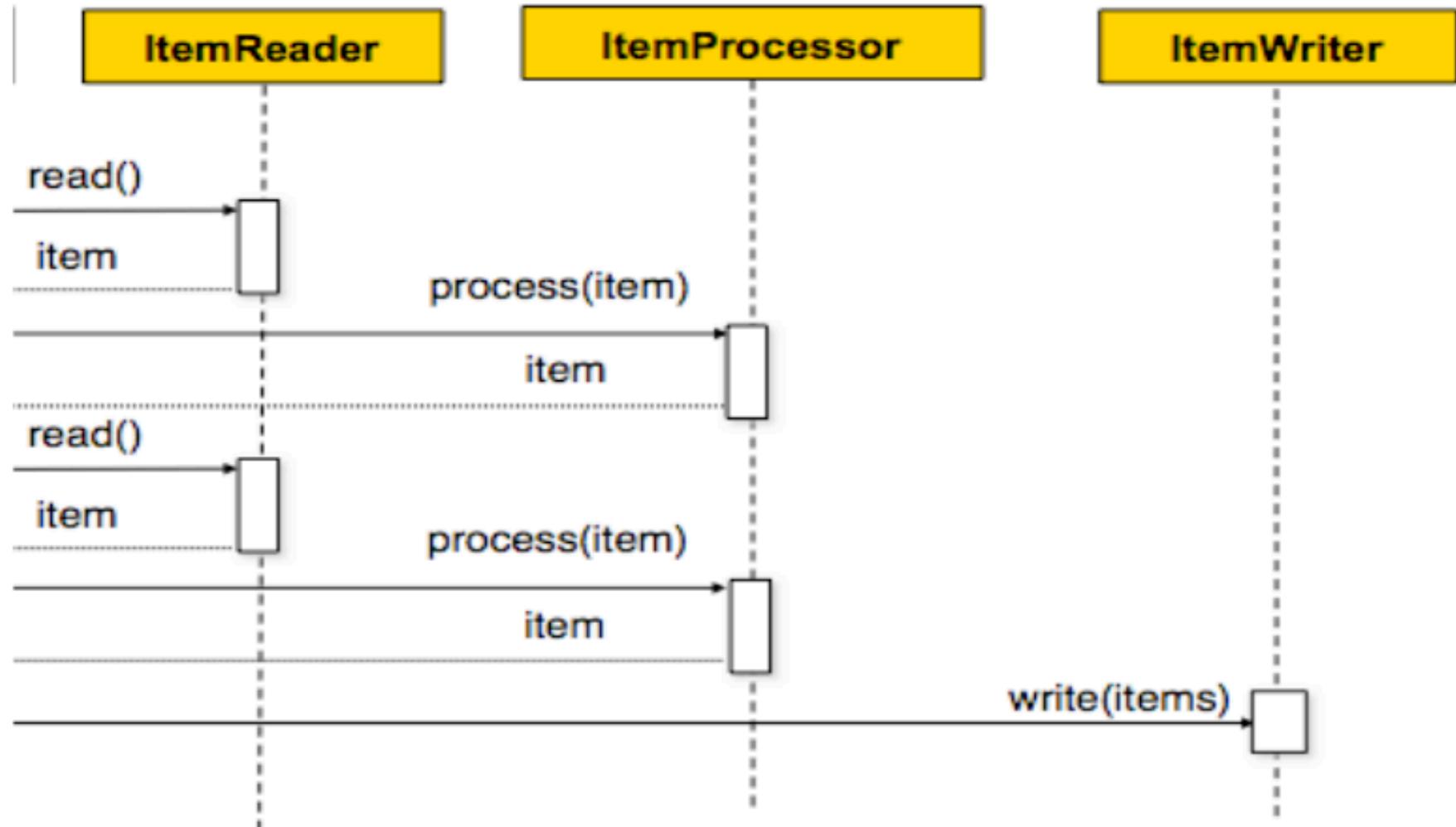


Chunk



- Besoin
 - Lire, transformer et écrire
- Chunk
 - Lit et transforme chaque recette successivement
 - Ecrit les recettes, par lot

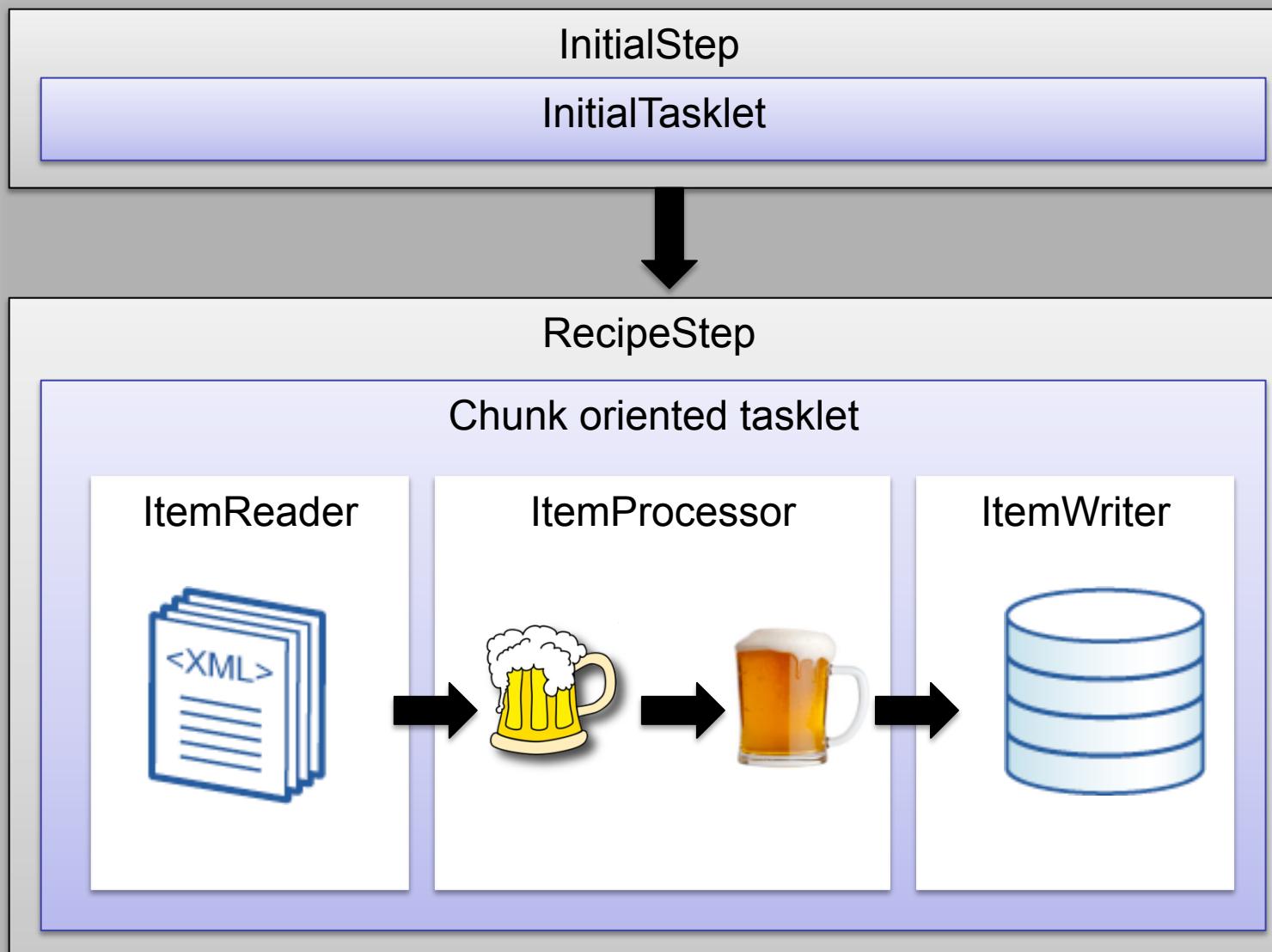
Chunk



Taille du lot = commit interval = 2



Step



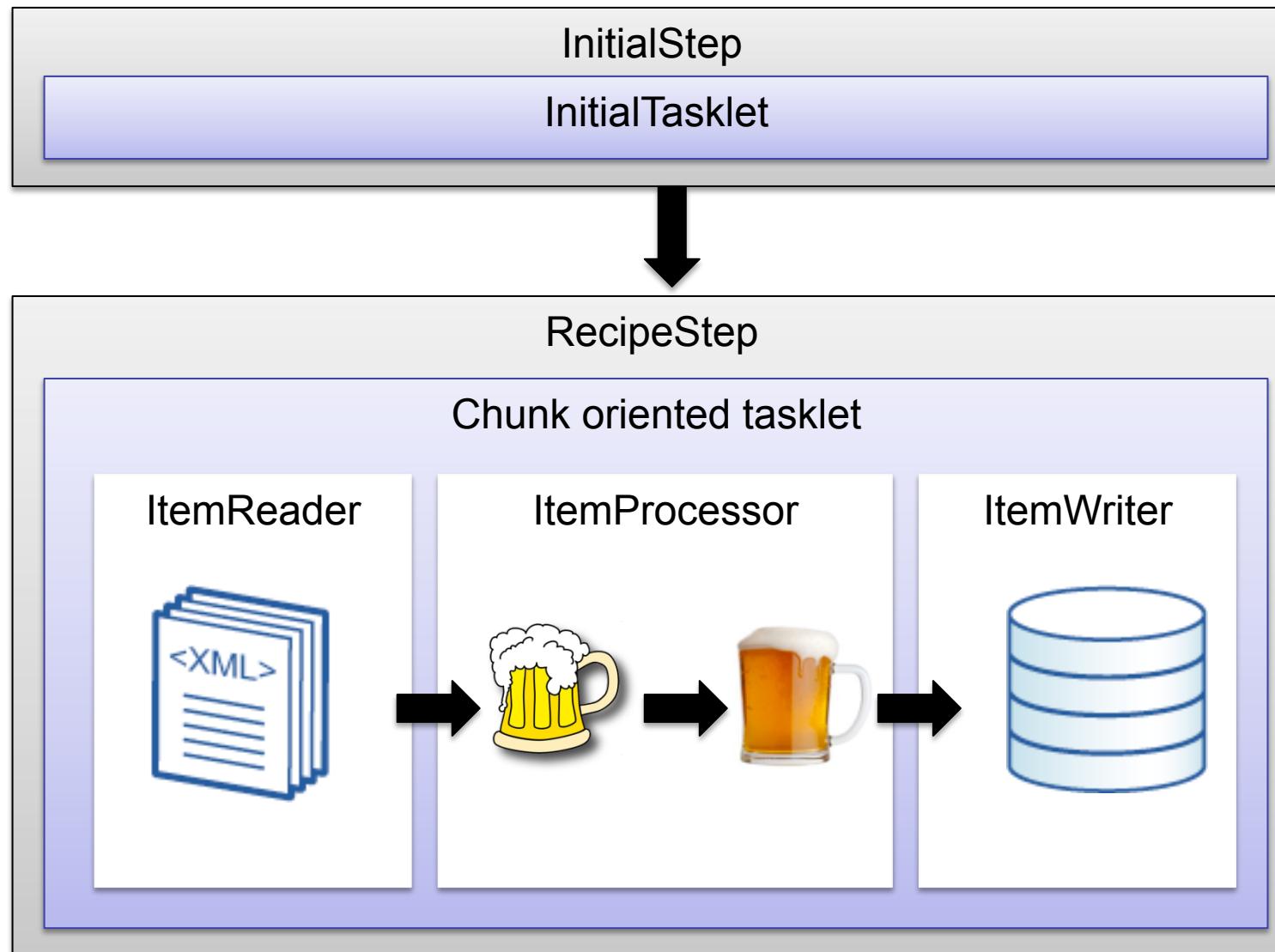
Step

- Besoin
 - Etape dans le processus du batch
 - Contrôle le workflow

```
<batch:step id="initialStep">
    <batch:tasklet ref="initialTasklet" />
    <batch:next on="*" to="recipeStep" />
</batch:step>
<batch:step id="recipeStep">
    <batch:tasklet>
</batch:step>
```

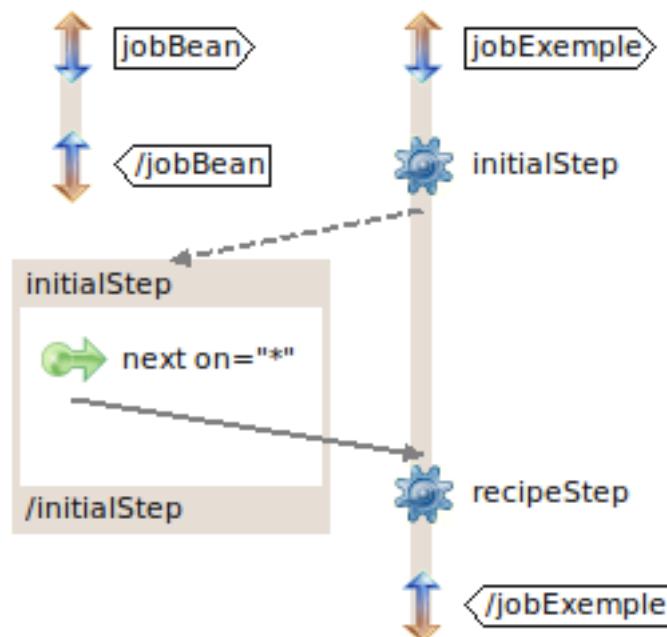


Job



Job

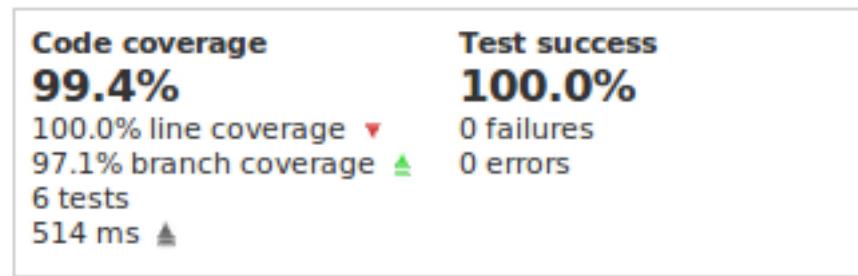
- Besoin
 - Décrire les étapes du batch
 - Composé d'un ou plusieurs steps



Spring Tools Suite



Tests automatisés



- Coder un batch en TDD
- Tests d'intégration facilités

<https://github.com/obazoud/spring-batch-sug>



Forces et faiblesses

- ✓ Batchs complexes mieux maintenables
- ✓ Productivité
- ✓ Tests

- ✗ Ticket d'entrée
- ✗ Ne convient pas à tous les batchs



Retour d'expérience Ekino



- Avec « Spring Batch »
 - Moins de code produit
 - Plus de tests unitaires & intégrations
 - Spring Batch s'occupe de la montée en charge
 - SQL/MyBatis
- Gain par rapport à un batch « legacy »
 - XML 100Mo + SQL; Gain : de 60 mn à 8mn
 - 10 fichiers 4Mo + SQL; Gain : de 10mn à 1mn30s
 - 20 fois rapide qu'un batch en ... PHP ;)



JSR-352 « javax.batch »



Ressemble beaucoup à
Spring Batch



Spring Batch

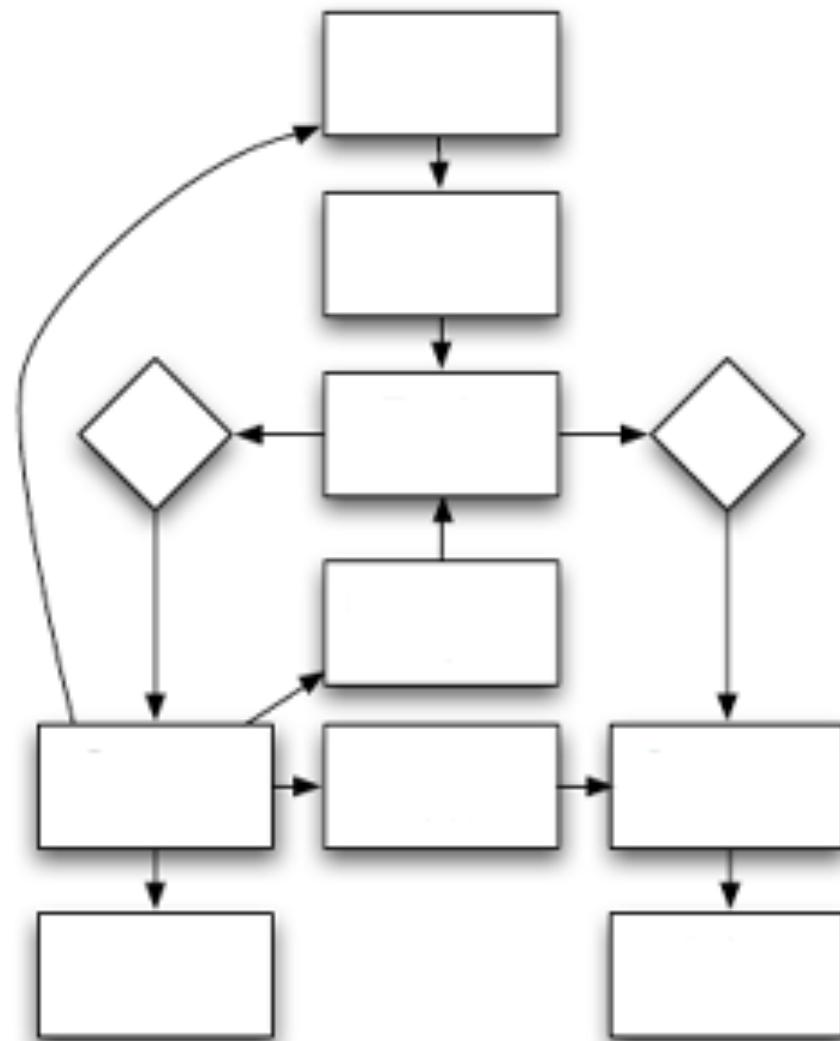
Allons plus loin



Un batch plus robuste



Flow



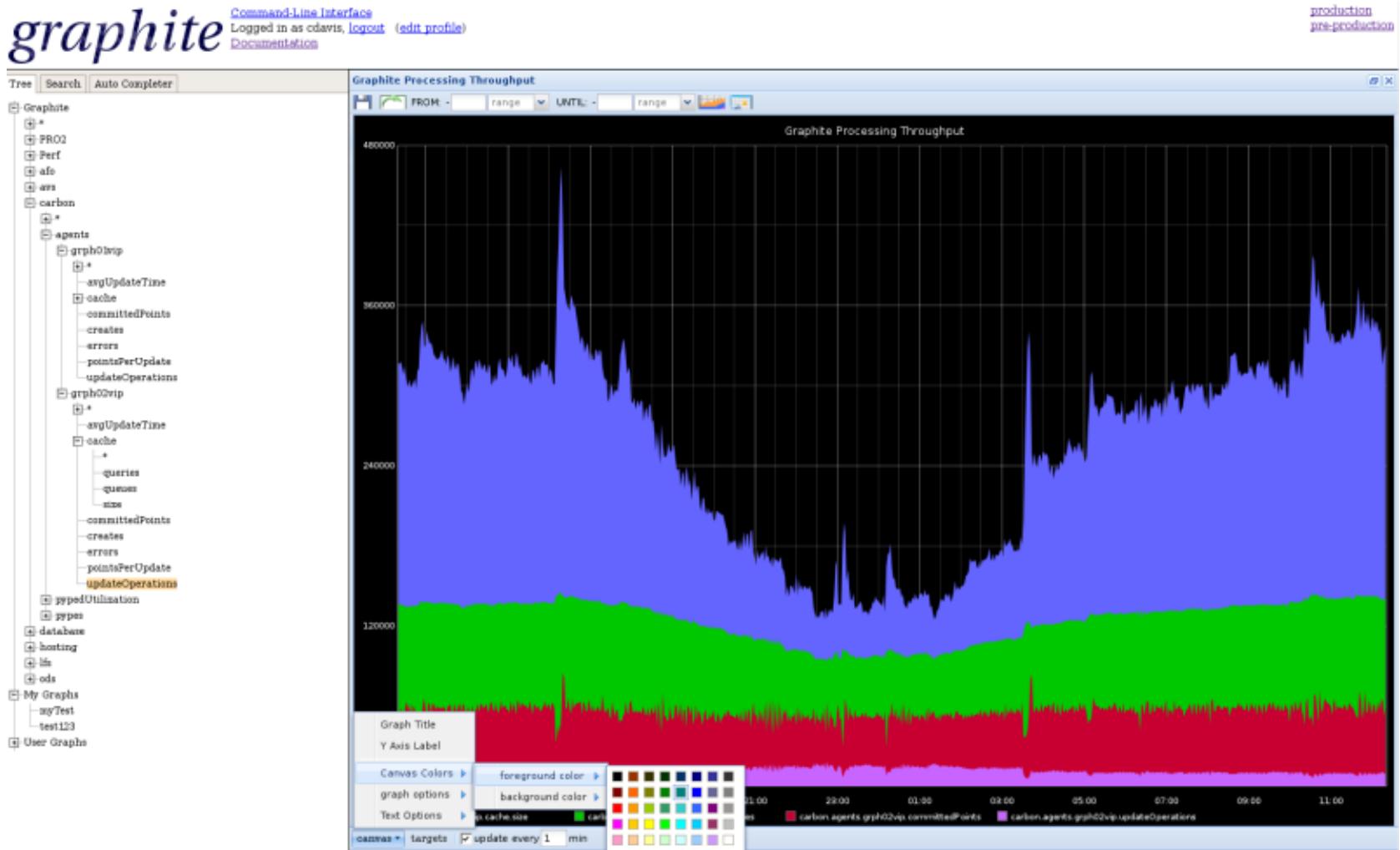
Scaling



<http://www.flickr.com/photos/claudiasofia99/2878579560/>



Monitoring



Spring Batch Admin

Spring Batch Admin



Home Jobs Executions Files SpringSource Spring Batch

Job Names Registered

Name	Description	Execution Count	Launchable	Incrementable
infinite	No description	0	true	false
job1	No description	0	true	true
job2	No description	0	true	false

Rows: 1-3 of 3 Page Size: 20

© Copyright 2009-2010 SpringSource. All Rights Reserved.

Contact SpringSource



Un batch plus robuste reprise sur erreurs



Reprise sur erreur

- Spring Batch « out-of-the-box » propose:
 - Sauter les erreurs non bloquantes (skip)
 - survivre à quelques données invalides
 - Recommencer un traitement (retry)
 - survivre à une indisponibilité temporaire
 - Redémarrer un batch (restart)
 - on est en mesure de le relancer



Reprise sur erreur: Skip

```
0001;ABC;DEF;  
0002;ABC;DEF;  
000zxjgxjdjghjsdfkud  
0004;ABC;DEF;
```

- Par défaut, au-delà du « skip-limit », failed

```
public interface SkipPolicy {  
    boolean shouldSkip(Throwable t, int skipCount) throws SkipLimitExceededException;  
}  
  
public interface SkipListener<T,S> extends StepListener {  
    void onSkipInRead(Throwable t);  
    void onSkipInWrite(S item, Throwable t);  
    void onSkipInProcess(T item, Throwable t);  
}
```



Reprise sur erreur: Retry



Reprise sur erreur: Retry

- Relancer une opération si indisponibilité
- Par défaut, au-delà du « retry-limit », failed
- Personnaliser le « retry »
 - RetryPolicy
 - Stratégie entre 2 « retry »: BackoffPolicy
- RetryListener pour écouter



Reprise sur erreur: Restart



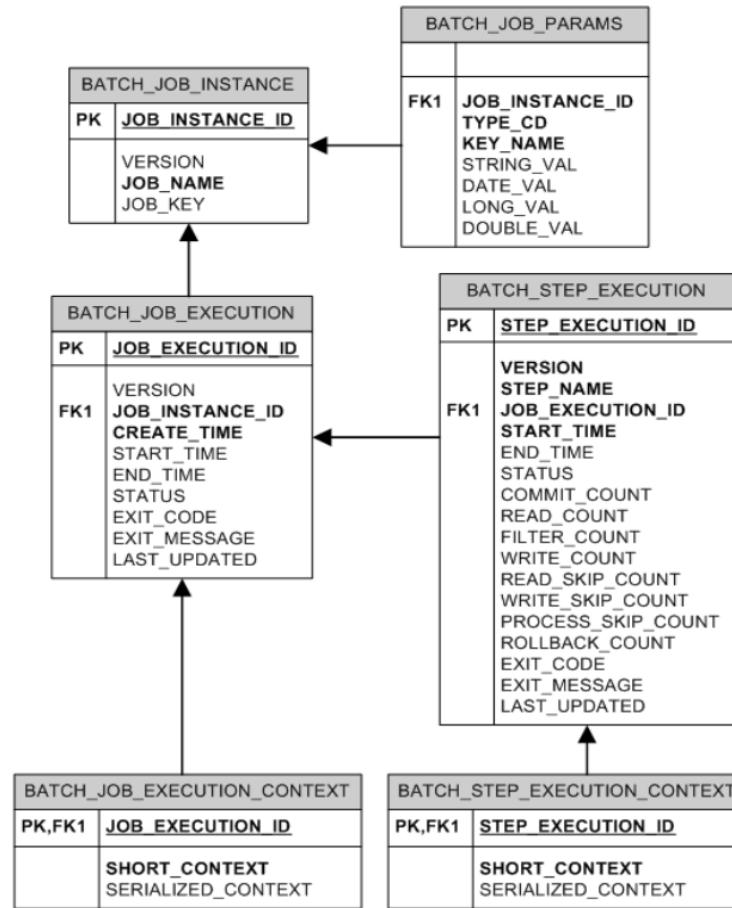
**“Whenever something goes wrong,
I just push this little button and restart.
I wish my whole life was like that!”**

joe-ks.com



Reprise sur erreur: Restart

- Spring batch stocke des méta data



Reprise sur erreur: Restart

- Indique si le job est « restartable »

```
<job restartable="true/false">  
  (...)  
</job>
```

- Indique le nombre max de « restart »

```
<job restartable="true">  
  <step id="stepA">  
    <tasklet start-limit="3">  
      (...)
```

- Reader/Writer doivent être « restartable »



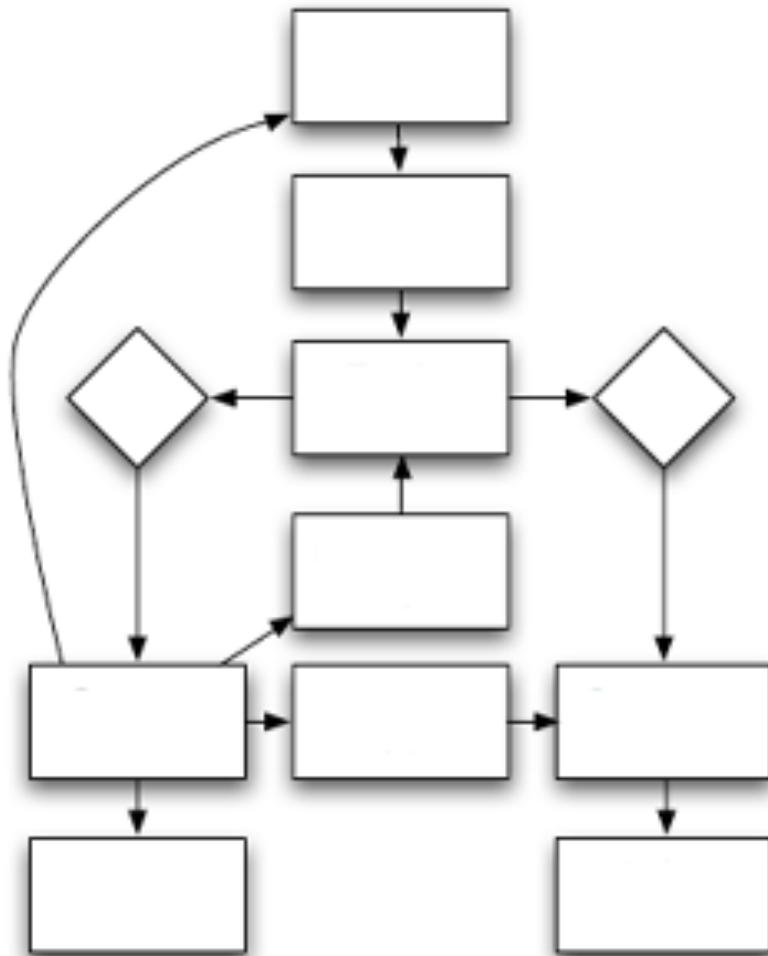
Reprise sur erreur

Conclusion

- Ecarter des items défectueux (skip)
- Recommencer une step s'il y a une erreur temporaire (retry)
- Redémarrer un job (restart)

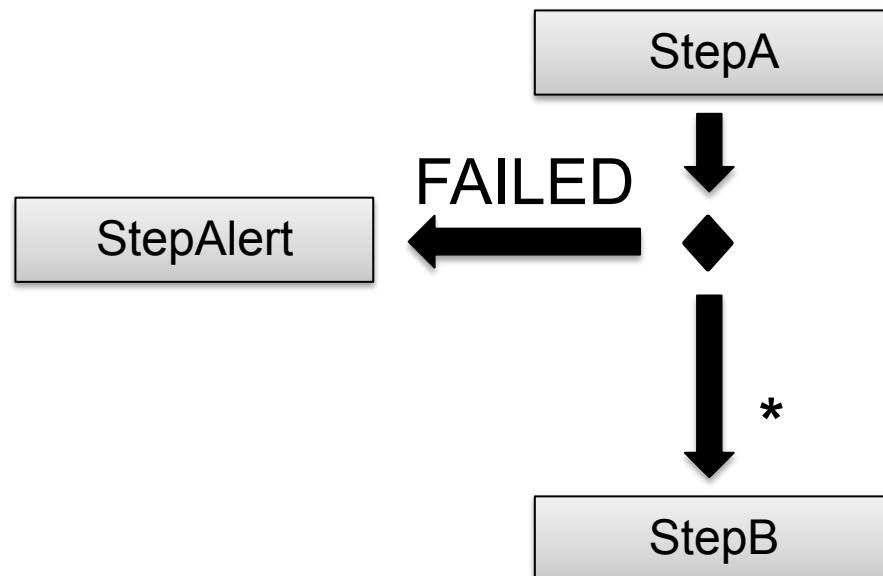


Flow



Gestion du flow

- Exemple plus simple ;)



Gestion du flow

```
<job id="jobExemple"
    <step id="stepA">
        (...)
        <next on="*" to="stepB" />
        <next on="FAILED" to="stepAlert" />
    </step>
    <step id="stepB">
        (...)
    </step>
    <step id="stepAlert">
        (...)
    </step>
</job>
```



Gestion du flow

- Status
 - Etat du job/step en cours/après d'exécution
- Créer et utiliser ses propres « status »
- Gérer le flow
 - StepExecutionListener
 - JobDecider



Scaling



<http://www.flickr.com/photos/claudiasofia99/2878579560/>



Scaling



Horizontal Scaling

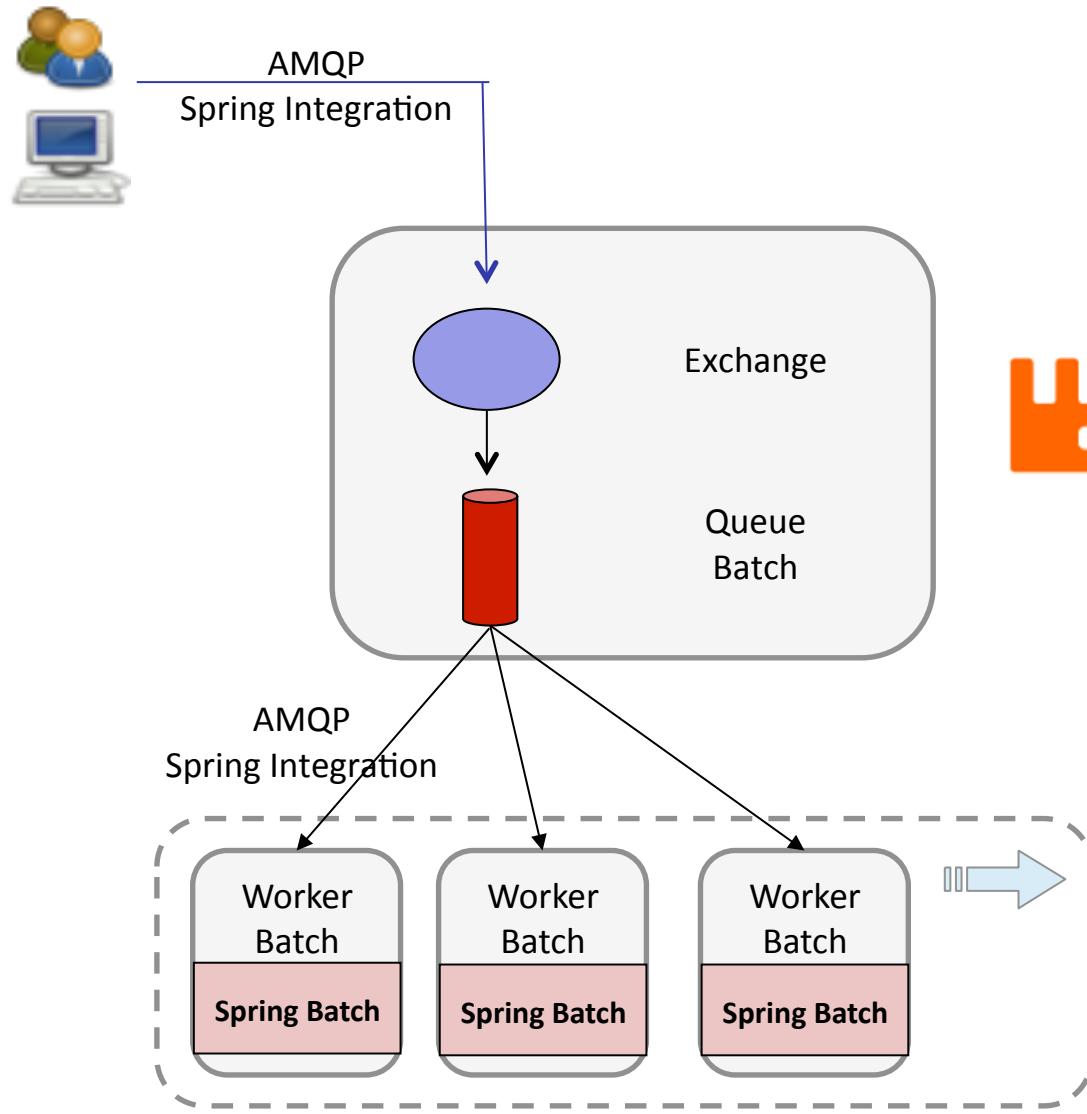
- RabbitMQ: Message-Oriented Middleware
 - Découpler le producteur des consommateurs



- Batch Orienté Tâche ou Worker



Horizontal Scaling



 **RabbitMQ**
Messaging that just works™



Scaling

- Si un ItemProcessor est « lent »
- ItemProcessor en asynchrone
- ItemWriter attend la fin des processors
- AsyncItemProcessor / AsyncItemWriter



Scaling

- Multi-threaded Step (Local)
 - Une Step est multithreaded
- Parallel Steps (Local)
 - Les Steps sont exécutées en parallèle



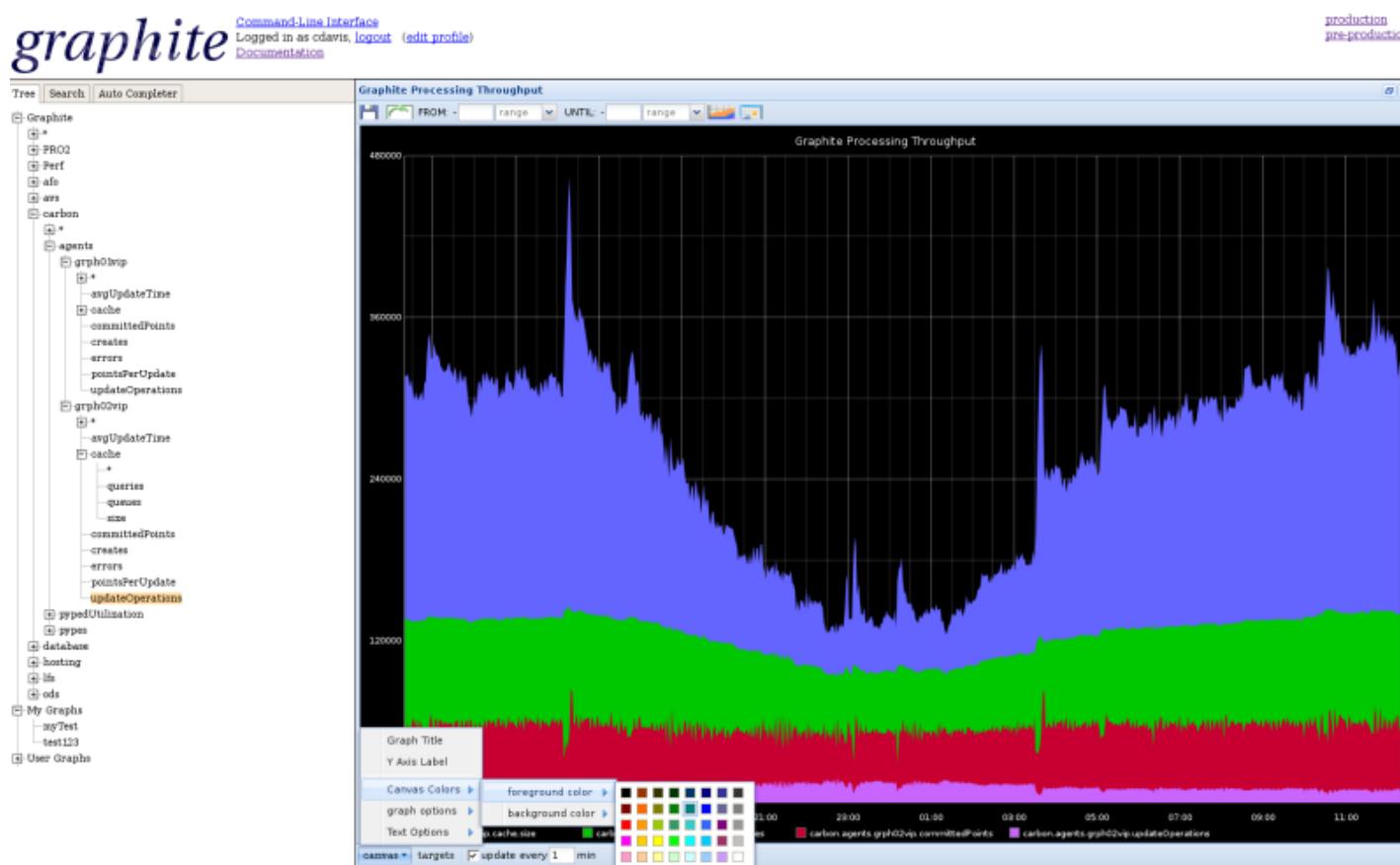
Scaling

- Remote Chunking of Step (Remote)
 - Distribution des chunks (Horizontal Scaling)
 - Reader sur les master
 - Processor/Writer sur les slaves
- Partitioning a Step (Local + Remote)
 - Découpe les données suivant une clé de partition



Monitoring

- Graphite <http://graphite.wikidot.com>



Monitoring

- Collectd
 - <http://collectd.org>
 - démon qui recueille périodiquement des statistiques de performance du système
- JMXTTrans
 - <https://github.com/jmxtrans/jmxtrans>
 - Interroge les données JMX de la JVM et les envoyer dans Graphite



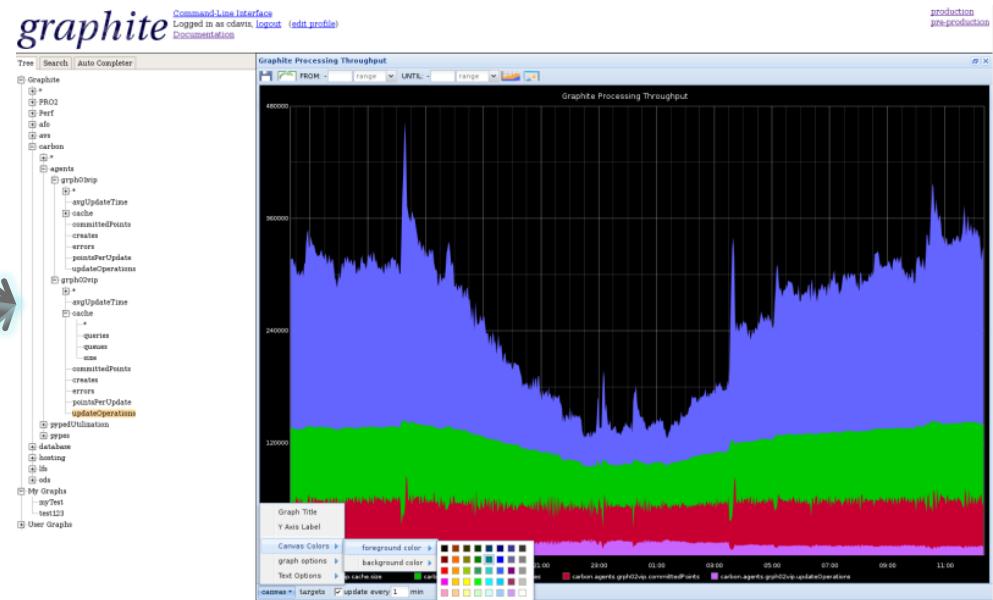
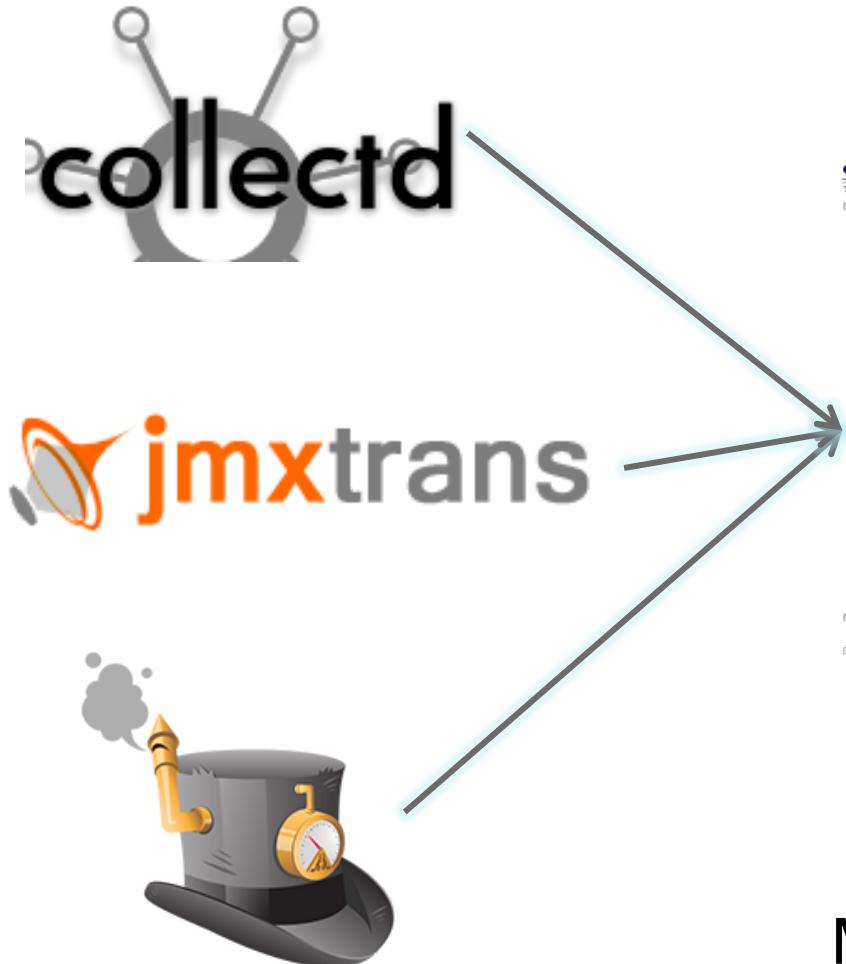
Monitoring



- Metrics
 - <http://metrics.codahale.com/>
 - Connaitre l'intérieur de votre application
 - Gauges, Counters, Meters, Histogram, Timers
- Extension pour Spring Batch
 - <https://github.com/obazoud/metrics-spring-batch>
 - Bientôt sur Maven Central



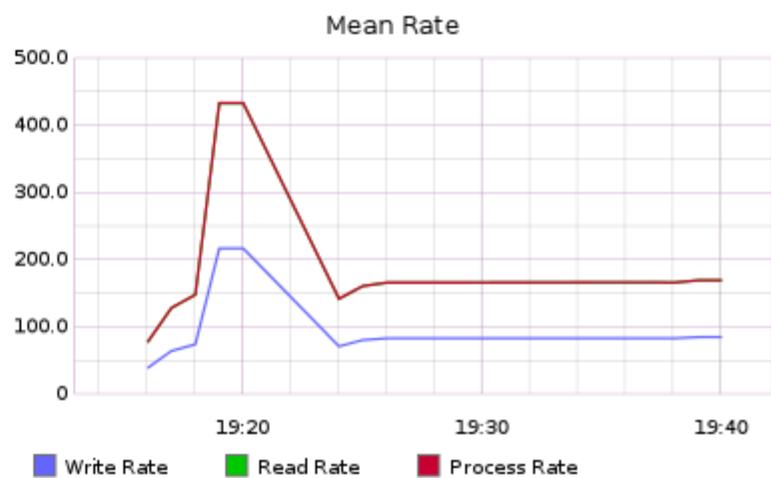
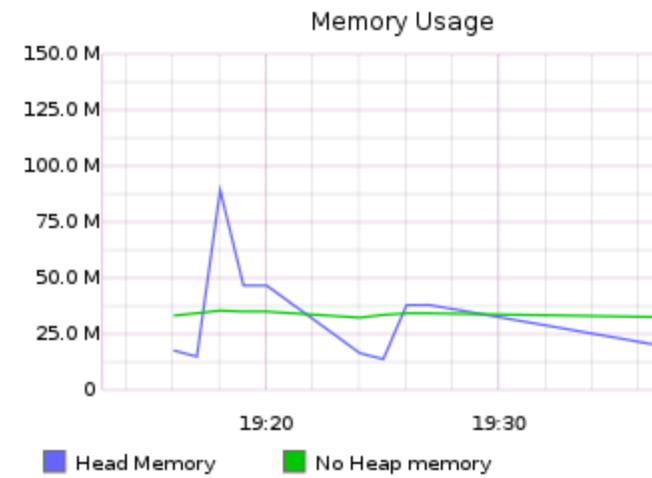
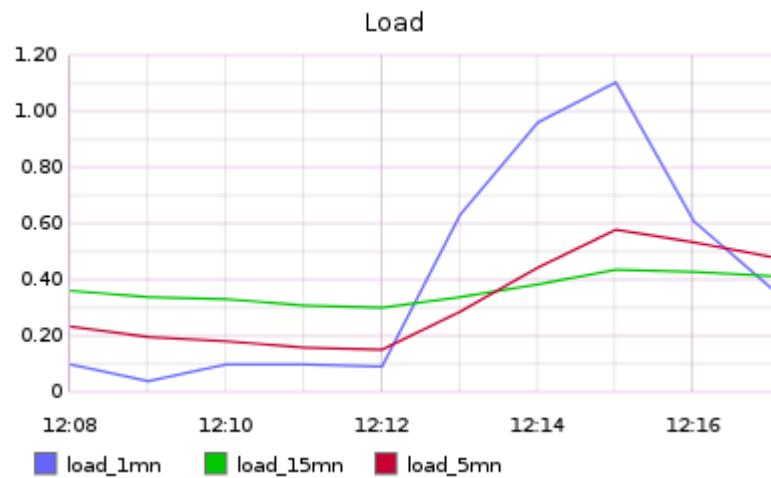
Monitoring



Mode push au lieu de pull



Monitoring



Spring Batch Admin

Spring Batch Admin



Home Jobs Executions Files SpringSource Spring Batch

Job Names Registered

Name	Description	Execution Count	Launchable	Incrementable
infinite	No description	0	true	false
job1	No description	0	true	true
job2	No description	0	true	false

Rows: 1-3 of 3 Page Size: 20

© Copyright 2009-2010 SpringSource. All Rights Reserved.

Contact SpringSource



Spring Batch Admin

- Console Web pour Spring Batch
 - « Standalone » ou « Embedded »
 - API Rest
 - Uploader une configuration, un fichier à traiter
 - « Customisable »
 - Informations sur les jobs, steps, ...



Liens

- Le code de la présentation
 - <https://github.com/obazoud/spring-batch-sug>
- Spring Batch 2.1.x
 - <http://static.springsource.org/spring-batch>
- Spring Batch 2.2 coming soon
- Articles sur le web
 - <http://www.theserverside.com/news/1363855/Spring-Batch-Overview>
 - <http://www.infoq.com/presentations/syer-introducing-spring-batch>
 - <http://blog.zenika.com/index.php?post/2010/03/05/To-batch-or-not-to-batch>
 - <http://blog.octo.com/spring-batch-par-quel-bout-le-prendre/>

