Cloudster

K-means algorithm for cloud computing

{stephane.caron, guillaume.claret, anisse.ismaili, jacques-henri.jourdan, michael.mathieu, mathieu.prevot, guillaume.seguin, yingjie.xu}@ens.fr

École Normale Supérieure - Department of Computer Science

May 20 2009

Cloudster ?

K-means algorithm

Goal : given N objects, optimally partition them into K clusters.
Basic algorithm :

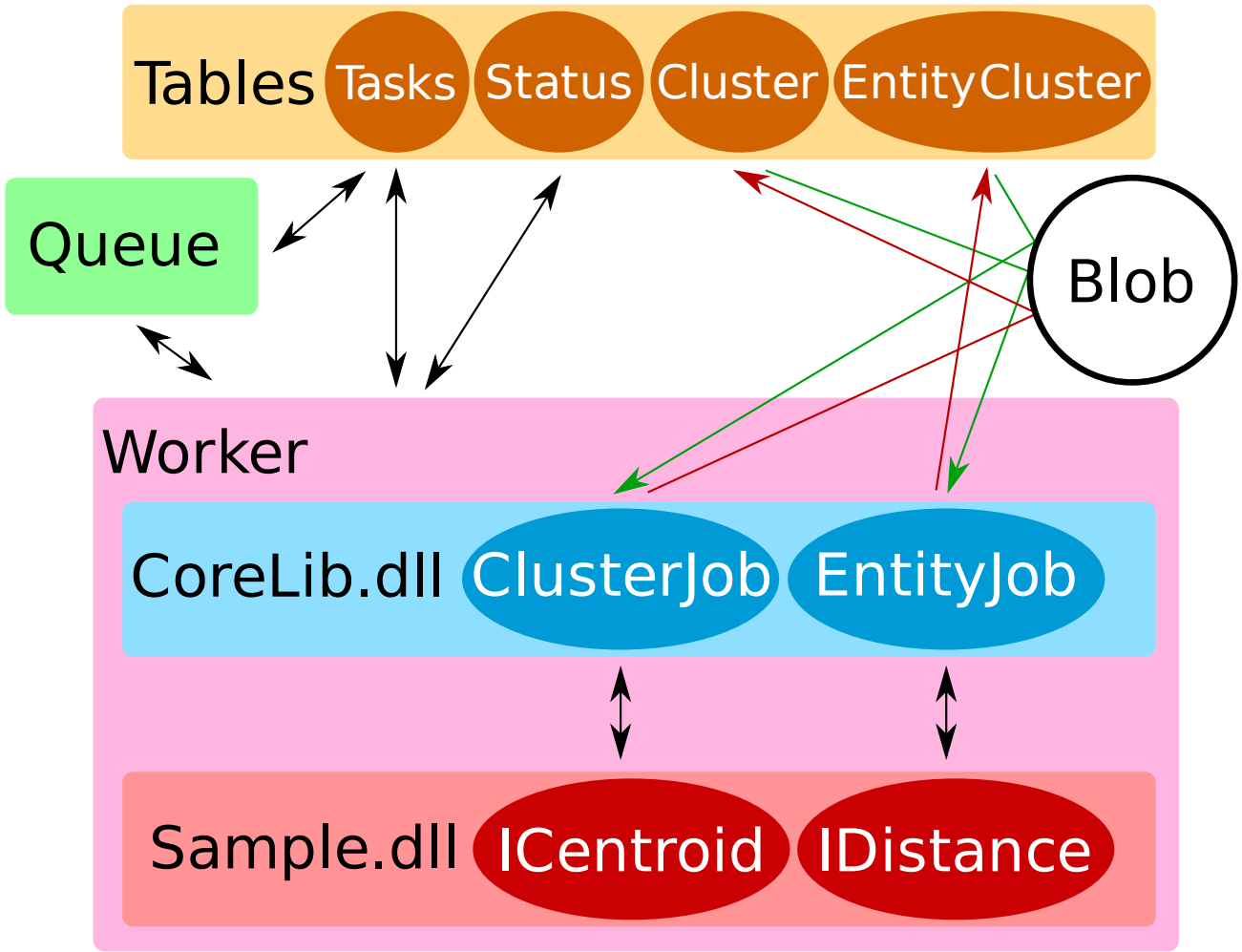
```
Randomly initialize groups
Iterate:
    foreach point p:
            Find nearest centroid C(p)
            Add p to the C(p) group
    Update centroids
```

About Cloudster

- Generic *k*-means algorithm implementation : feel free to feed it with your distance & centroïd computation functions !
- Heavily scalable : uses Windows® Azure cloud-computing platform
- Written in C# & uses the .NET framework
- BSD licensed, open development @ `http://cloudster.sourceforge.net`

Design

Design


Tables
Tasks
Status
Cluster
EntityCluster
Queue
Blob
Worker
CoreLib.dll
ClusterJob
EntityJob
Sample.dll
ICentroid
IDistance

User interaction

The CLI way

Three separate tools :

- The **Builder**, which initializes the blob storage and tables and uploads the initial entities
- The **Tester**, which starts the algorithm (either the sequential one or the cloud computed one)
- The **Evaluator**, which computes the *score* of the current algorithm state

The web way

A remote web interface, using Azure's web roles power.
Prefered way for interacting with the cloud : easier, better, faster :

- Unifies the CLI tools into a single interface
- Enables thorough monitoring of algorithm state (tasks, results)
- Enables case-specific visualisations of algorithm results

Future enhancements

Future (possible) core features

- Use reflection to unify involved tools
- Blob storage handling improvements :
  - Assign workers to specific groups of entities
  - Improve entities cache
  - Store multiple entities in each blob
- Split computations and storage queries to dedicated threads
- Enable the user to add/remove entities on the fly
- Table repair tool

Upcoming samples

- Sparse vectors sample
- Image comparison sample, based on GIST algorithm (currently investigating some implementation bugs)
- DNA sequences comparison sample, based on NAligner, using FASTA file format

Questions ?