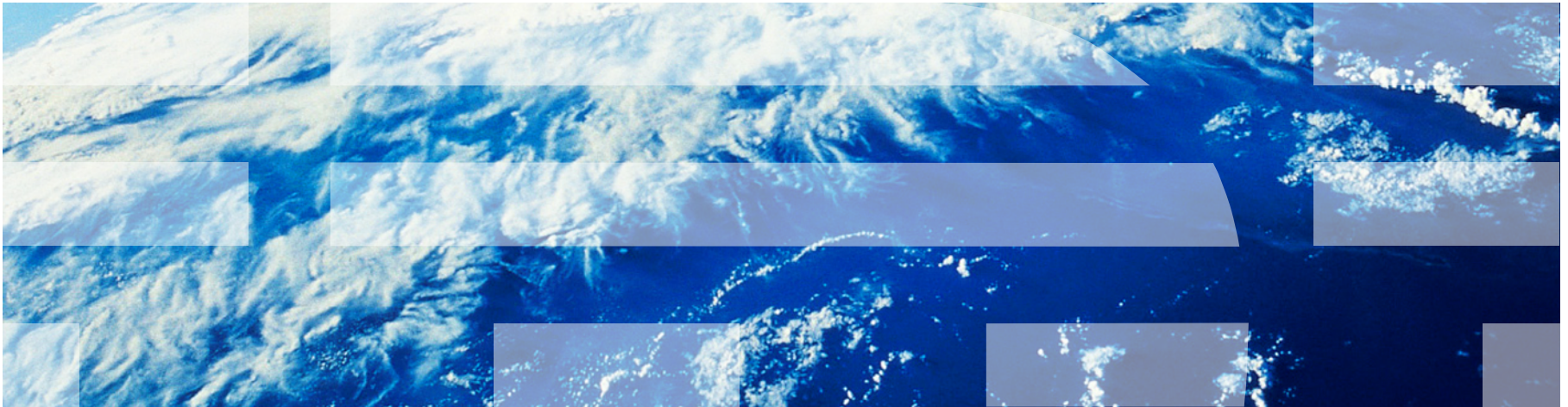

UIMA / DUCCE update, 2013

Marshall Schor, IBM Research



UIMA Background

- UIMA – Apache Open Source framework allowing assembly of independently developed Annotators in pipelines
 - Widely used both in Industry, Government, and Academia
 - Key underpinning of IBM’s “Watson” machine
 - Goal – build community around advancing analysis of unstructured data

- UIMA-AS – adds queue-based scale-out capabilities to exploit multi-core and multi-machine (in clusters)
 - Open Source – part of Apache UIMA

- (new, partially funded by Sharp): UIMA-DUCC – adds Distributed UIMA Cluster Controller to manage large clusters of compute nodes running dynamically-scaled-out UIMA pipelines

Before DUCC – some (not-so-hypothetical) issues

- Group acquires computing cluster, has long-running discussion on how to exploit it
 - Often ends up with various people laying claim to some subpart of the cluster for their experiments.
 - Dynamic reallocation of resources to existing Job needs not done – some jobs run slower, others can't fully use their allocated resources.
 - Machines with 8 cores are under-used because it's too hard to figure out how to make use of the other cores.
- As cluster grows, simple tasks like finding and reviewing all the logs for a particular scaled-out run become more tedious
- Responses to system failures (HW or SW) ad-hoc, often requiring redoing entire “job”
- Running too many “jobs” on the same machine (imagine it has 8 cores) sometimes over-commits the RAM, resulting in very slow execution due to swapping
- Complex UIMA pipelines often make use of shared services
 - These have to be manually started and kept up for multiple jobs.
 - Jobs fail if the service is not available for some reason

After DUCC

- SysAdmins define the Cluster (can be a mix of different kinds of machines)
- Users define their Jobs, specifying parameters of particular importance for UIMA annotators (such as how much RAM they need).
- There's a standard (set of) supported way(s) to describe scaling a pipe line over data.
 - Typically involves running a pipeline on multiple instances of data, collecting the results
 - Can be scaled by adding more pipeline instances
 - Can be restarted (if failure) on failing instance of data
- Users submit Jobs to the Cluster
 - Cluster dynamically allocates machines and spreads work, using fair-share allocation, done dynamically. For example:
 - if only 1 Job, it gets all the resources
 - 2nd job arrives, 1/2 the resources have their running processes for Job 1 “killed” (they’ll recover, running on the reduced resources available to Job1) and given to new Job2.
 - RAM resources managed to avoid overcommitting (avoiding swapping).
- Services managed
 - Needed ones started/recovered, at specified replication (scale)
 - Jobs held until their services are available

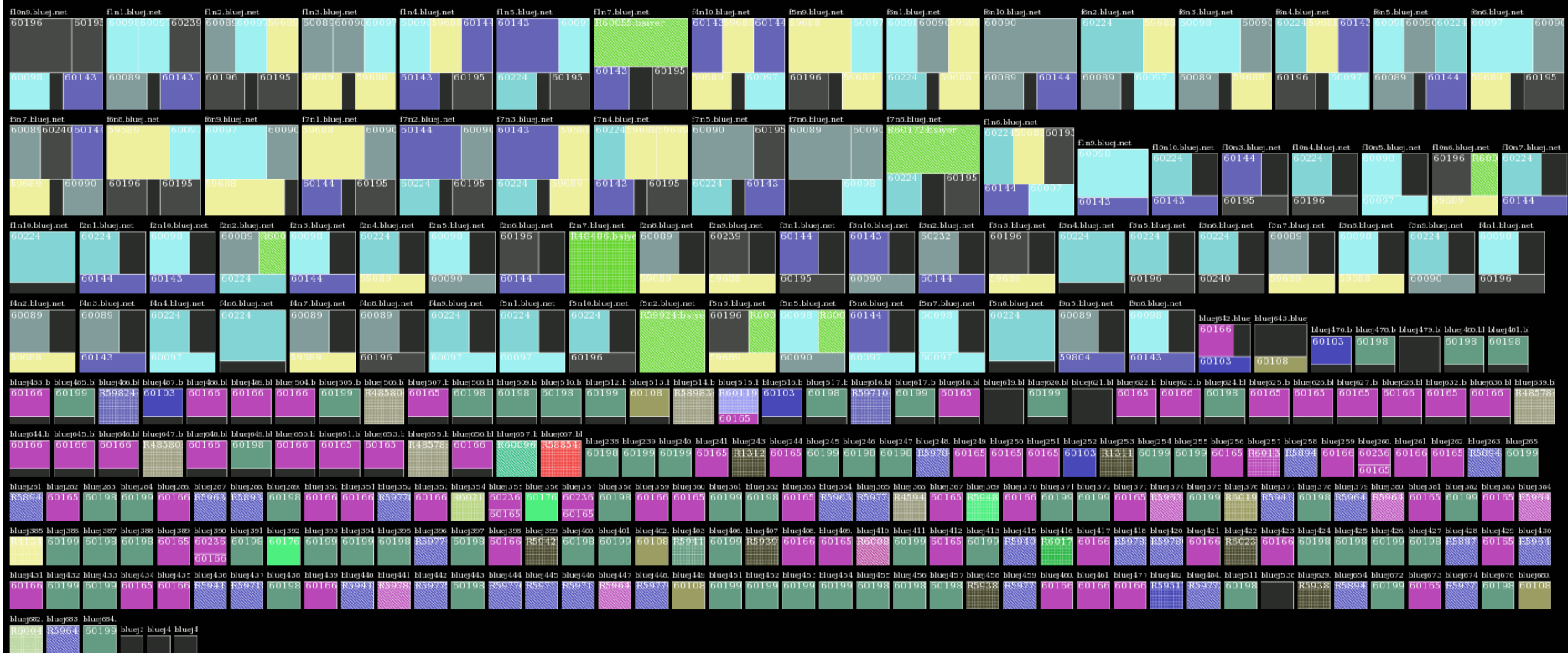
Monitoring, Visibility

- Web-based aggregations with drill-down for detail
- Machines + what they're running – each “user” has different color, grey = not allocated
- Size ~ memory (the most constrained resource)

DUCC nodes

[all jobs](#) [job flows](#) [history](#)

Total: 1282 shares of 15GB (87% used)
 Shares for jobs: 949, reserves: 69, services: 107



Jobs

- Browser Web GUI provides monitoring and control
- One Panel lists all the jobs
 - Filter by user etc.
 - Drill-down into machines/processes where that job is scaled out
 - One click access to logs, to start monitoring console (jconsole), etc., for each part
 - See how far the Job is through the set of items to process
- Jobs run with resource constraints for the user
 - (e.g., disk quota, r/w permissions)
- Performance aggregation per process, over all work-items
 - Breakdown by UIMA Components
 - Average (per work-item), minimum and maximum times (to catch outliers)
- Auto Retry work-item on errors
- UIMA-specific features
 - Delay scale-up until UIMA initialize phase proves successful
 - Delay resource allocation to Job until all services it depends on are available

DUCC Status

- In daily use at IBM Research, managing the cluster used for Watson machine Research
- Contributed to Apache UIMA, marching toward first release
 - Before year-end 2013