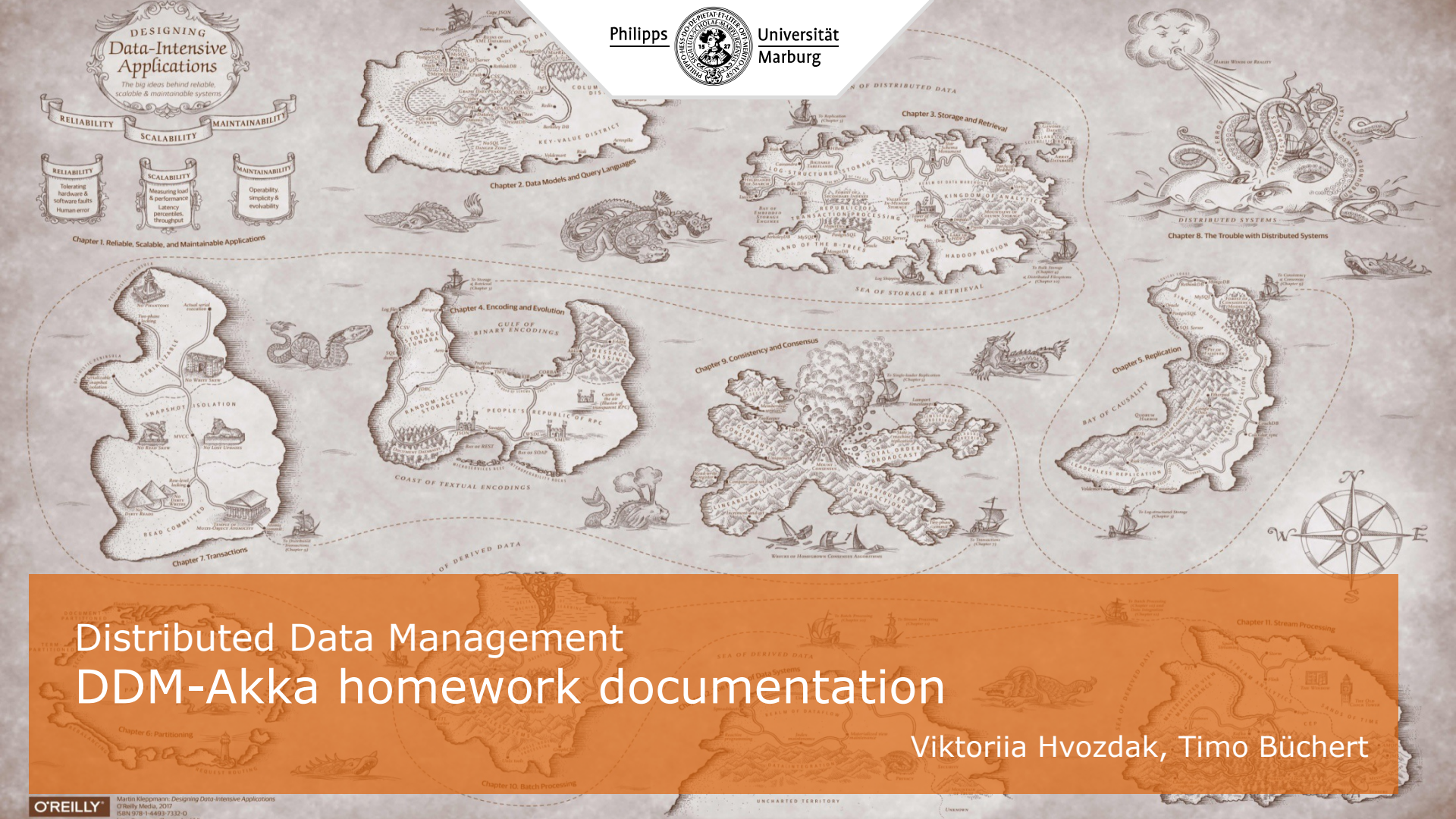

Philipps Universität Marburg

Distributed Data Management

DDM-Akka homework documentation

Viktoriia Hvozdak, Timo Büchert

O'REILLY
Martin Kleppmann: *Designing Data-Intensive Applications*
O'Reilly Media, 2017
ISBN 978-1-4493-7332-0

DDM-Akka homework documentation

Responsibility of the main actors

Philipps Universität Marburg

- DependencyMiner
  - Collects all batch messages from the input reader and checks completion of input reading.
  - Creates candidates for unary inclusion dependencies (INDs) and stores them until send to a dependency worker for verification (A candidate contains the table ID, attribute name and the data for the respective LHS and RHS column).
  - Keeps track of all dependency workers and assigns tasks if a new worker joins or becomes available after finishing its current task.
  - Notifies the result collector after completion of all tasks.
  - The task (including the necessary data) is sent to the dependency worker via large message proxy.
- DependencyWorker
  - Receives task messages from the dependency miner and validates the INDs by constructing a HashSet of the RHS and a HashSet of the LHS values. This way the dependency worker can easily check whether the distinct RHS values contain all LHS values while the HashSet automatically takes care of possible duplicates in the data.

Note: the submitted solution works locally on a MacBook Pro with 16GB Ram and a default max heap size of the jvm of 4294967296 bytes. It produces meaningful dependencies both standalone and with several locally started worker processes. Sadly, our solution does not work with a limited heap size of 2 gigabytes. We tried various approaches (e.g. sending the data to the workers separated from the tasks), but almost all the memory on the master was already occupied after the inputs were read – around 80 MB left in the heap was not enough to continue working.