1. Record Nr. UNINA9910485028003321

Autore	Wu Huijun (Writer on cloud computing)
Titolo	Heron streaming : fundamentals, applications, operations, and insights / / Huijun Wu, Maosong Fu
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2021] ©2021
ISBN	3-030-60094-7
Descrizione fisica	1 online resource (211 pages) : illustrations
Disciplina	004.3
Soggetti	Information retrieval Data mining Big data Information organization Recuperació de la informació Mineria de dades Dades massives Llibres electrònics
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Foreword -- Preface -- Who This Book Is For -- How This Book Is Organized -- What You Need for This Book -- Typographical Conventions -- Acknowledgments -- Contents -- About the Authors -- Part I Heron Fundamentals -- 1 Stream Processing -- 1.1 Big Data Processing -- 1.1.1 Lambda Architecture -- 1.1.1.1 Batch Processing Layer -- 1.1.1.2 Stream Processing Layer -- 1.1.1.3 Serving Layer -- 1.1.2 Kappa Architecture -- 1.2 Big Data Stream Processing -- 1.3 From Apache Storm to Apache Heron (Incubating) -- 1.3.1 Motivation for Heron -- 1.3.2 Heron Design Goals -- 1.3.3 Join the Apache Heron (Incubating) Community -- 1.4 Stream Processing Tools -- 1.5 Summary -- References -- 2 Heron Basics -- 2.1 Topology Data Model -- 2.1.1 Topology -- 2.1.2 Spout -- 2.1.3 Bolt -- 2.1.4 Grouping -- 2.2 Heron Architecture and Components -- 2.2.1 Cluster-Level Components (Six Components) -- 2.2.1.1 Scheduler -- 2.2.1.2 State Manager -- 2.2.1.3 Uploader -- 2.2.1.4 Heron CLI -- 2.2.1.5 Heron

Tracker -- 2.2.1.6 Heron UI -- 2.2.2 Topology-Level Components (Four Components) -- 2.2.2.1 Heron Instance -- 2.2.2.2 Stream Manager -- 2.2.2.3 Topology Master -- 2.2.2.4 Metrics Manager -- 2.3 Submission Process and Failure Handling -- 2.4 Submit the First Topology -- 2.4.1 Preparation -- 2.4.2 Install the Heron Client -- 2.4.3 Heron Example Topologies -- 2.4.4 Submit the Topology JAR File -- 2.4.5 Observe the Running Topology -- 2.5 Summary -- References -- 3 Study Heron Code -- 3.1 Code Languages -- 3.2 Requirements for Compiling -- 3.3 Prepare the Compiling Environment -- 3.4 Source Organization -- 3.4.1 Directory Organization -- 3.4.2 Bazel Perspective -- 3.5 Compile Heron -- 3.6 Examine Compiling Results -- 3.6.1 Examine the API -- 3.6.2 Examine Packages -- 3.7 Run Tests -- 3.7.1 Unit Test -- 3.7.2 Integration Test -- 3.8 Summary -- References.
Part II Write Heron Topologies -- 4 Migrate Storm Topology to Heron -- 4.1 Prepare the Storm Topology Code -- 4.1.1 Examine the Storm Topology Code -- 4.1.2 Examine the Storm Flux Code -- 4.2 Migrate the Storm Topology Code to a HeronTopology Project -- 4.2.1 Adjust the Topology Java Code -- 4.2.2 Adjust the Project File pom.xml -- 4.2.2.1 Add Dependency -- 4.2.2.2 Build with Dependencies -- 4.2.3 Compile the Topology JAR File -- 4.3 Migrate Storm Flux to Heron ECO -- 4.4 Summary -- References -- 5 Write Topology Code -- 5.1 Before Writing Code -- 5.1.1 Design Topology -- 5.1.2 Choose a Heron API -- 5.2 Write Topology in Java -- 5.2.1 Code the Topology -- 5.2.1.1 Code Main -- 5.2.1.2 Code Spout -- 5.2.1.3 Code Bolt -- 5.2.2 Understand Tuple Flow -- 5.2.2.1 How Tuple Is Constructed -- 5.2.2.2 How Tuple Is Routed -- 5.3 Write Topology in Python -- 5.3.1 Code Main -- 5.3.2 Code Spout -- 5.3.3 Code Bolt -- 5.3.4 Compile and Run -- 5.4 Summary -- Reference -- 6 Heron Topology Features -- 6.1 Delivery Semantics -- 6.1.1 At-Least-Once -- 6.1.2 Effectively-Once -- 6.1.2.1 Requirements for Effectively-Once -- 6.1.2.2 Exactly-Once Versus Effectively-Once -- 6.1.2.3 Stateful Topologies -- 6.1.2.4 Implement Effectively-Once -- 6.2 Windowing -- 6.2.1 Windowing Concepts -- 6.2.2 Windowing Example -- 6.3 Summary -- Reference -- 7 Heron Streamlet API -- 7.1 Streamlet API Concepts -- 7.1.1 Streamlets -- 7.1.2 Operations -- 7.2 Write a Processing Graph with the Java Streamlet API -- 7.2.1 Sources -- 7.2.2 Sinks -- 7.2.3 Transform: Filter, Map, FlatMap -- 7.2.4 Partitioning -- 7.2.5 Clone and Union -- 7.2.6 Reduce by Key and Window -- 7.2.7 Join -- 7.2.8 Configuration -- 7.3 Write a Processing Graph with the Python Streamlet API -- 7.3.1 Source Generator -- 7.3.2 Processing Graph Construction -- 7.4 Write a Processing Graph with the Scala Streamlet API.
7.4.1 Install sbt -- 7.4.2 Source Directory -- 7.4.3 Compose Processing Graph -- 7.4.4 Examine the JAR File -- 7.5 Summary -- References -- Part III Operate Heron Clusters -- 8 Manage a Topology -- 8.1 Install Heron Client -- 8.1.1 What Is Inside heron-core.tar.gz -- 8.1.2 YAML Configuration -- 8.2 Run Topology -- 8.2.1 Topology Life Cycle -- 8.2.2 Submit Topology -- 8.2.3 Observe the Topology Running Status -- 8.3 Explore the ``heron'' Command -- 8.3.1 Common Arguments and Optional Flags -- 8.3.2 Explore ``heron submit'' Options -- 8.3.3 Kill Topology -- 8.3.4 Activate and Deactivate Topology -- 8.3.5 Restart Topology -- 8.3.6 Update Topology -- 8.4 Summary -- References -- 9 Manage Multiple Topologies -- 9.1 Install Heron Tools -- 9.2 Heron Tracker -- 9.3 Heron UI -- 9.4 Heron Explorer -- 9.5 Summary -- Reference -- Part IV Heron Insights -- 10 Explore Heron -- 10.1 Heron Processes -- 10.1.1 Java Processes -- 10.1.2 C++ Processes -- 10.1.3 Python Processes -- 10.2 State Manager -- 10.3 Heron Scheduler -- 10.3.1 Three Plans: Logical, Packing, and Physical -- 10.3.2 Restart Dead Processes -- 10.4 Data Flow -- 10.4.1 Capture