



REEF

DEVIEW
2015

딥러닝에서 스트림처리까지: 빅데이터 분석 메타 프레임워크 Apache REEF

2015년 9월 14일

전병곤
서울대학교 컴퓨터공학부

REEF

- A meta-framework that eases the development of Big Data applications atop resource managers such as YARN and Mesos
- Apache Incubator project since August 12, 2014
- 24 committers – committers from SNU, Microsoft, Alibaba, Purestorage, UCLA, UC Berkeley, UW, SKT
- Release 0.12.0-incubating on August 15, 2015

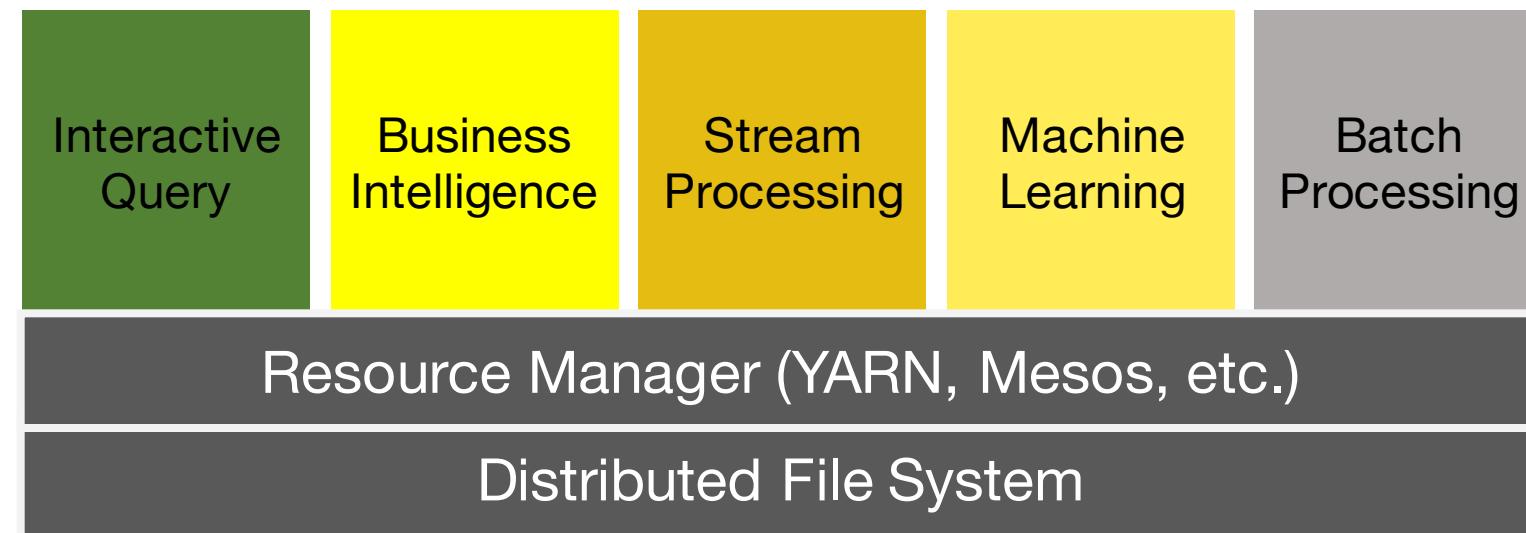
SNU CMSLab Collaborators

- Brian Cho
- Beom-Yeol Jeon
- Joo-Seong Jeong
- Dong-Jun Lee
- Yun-Seong Lee
- Young-Seok Yang
- Gyeong-In Yu
- Young-Bin Bok
- Geon-Woo Kim
- Tae-Hun Kim
- Gye-Won Lee
- Woo-Yeon Lee
- Tae-Geon Um

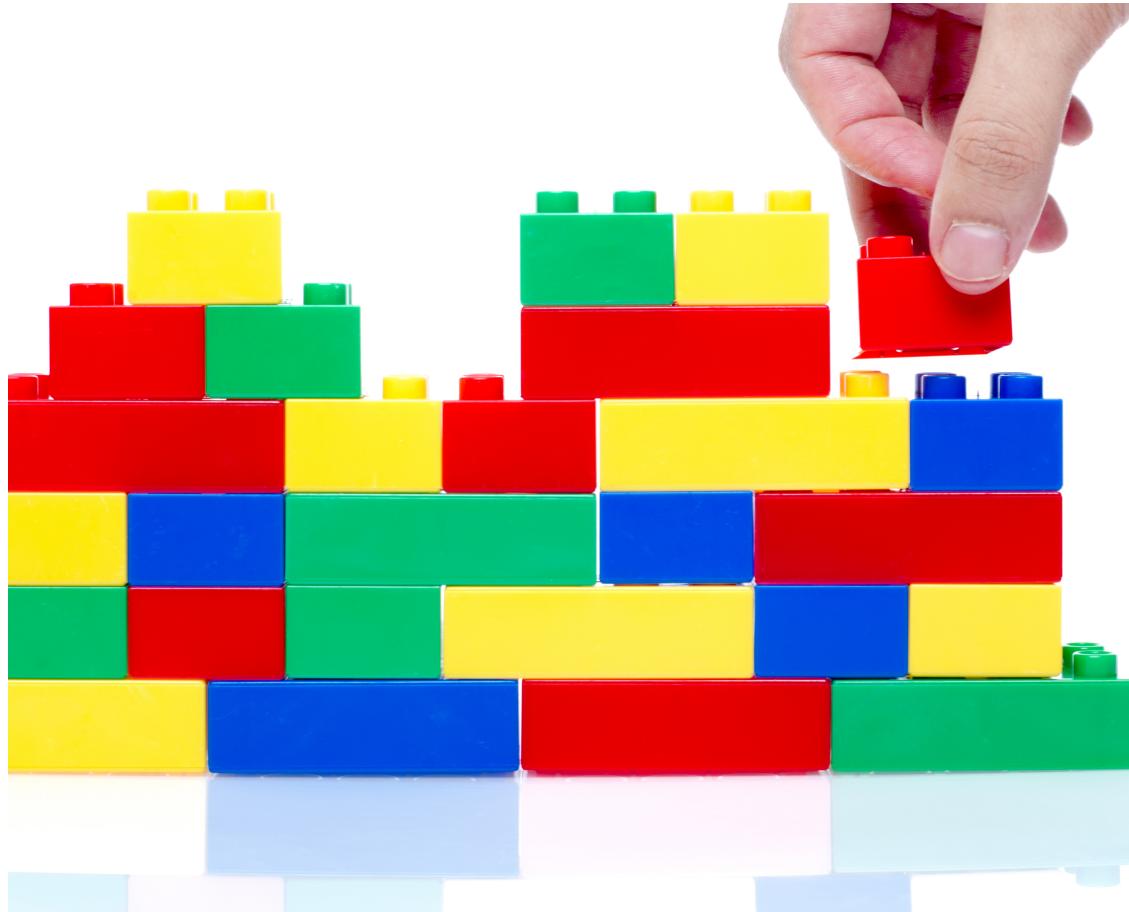
Roadmap

- What is REEF?
- Why would you want REEF?
- From REEF to CAY

Current Big Data Stack



Vision: Lego-style Big Data Analytics Creation



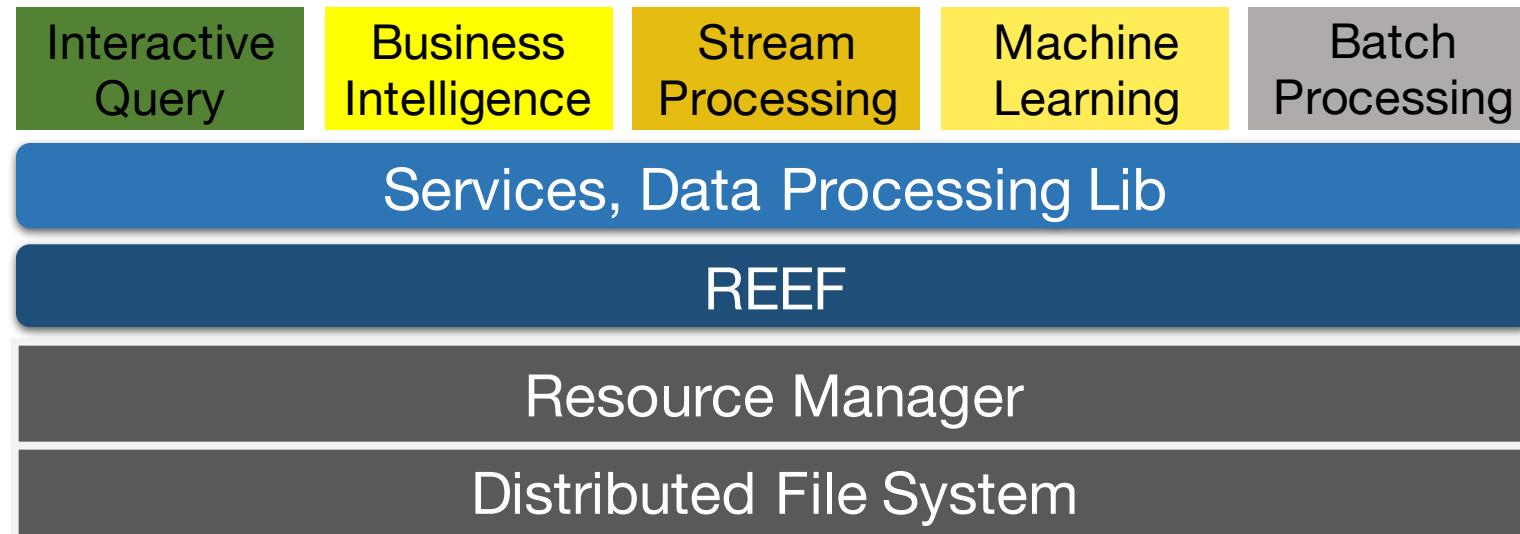
Decomposing Big Data Applications

- **Control plane:** coordinates application job tasks, handles faults, provides heartbeats, configures jobs, etc.
- **Data plane:** moves and processes data

Common Patterns

- A centralized per-job scheduler that observes the runtime state and assigns tasks to resources
- A runtime for executing compute tasks and retaining state in an organized fashion
- Communication channels for monitoring status and sending control messages
- Configuration management for passing parameters and binding application interfaces to runtime implementation

REEF

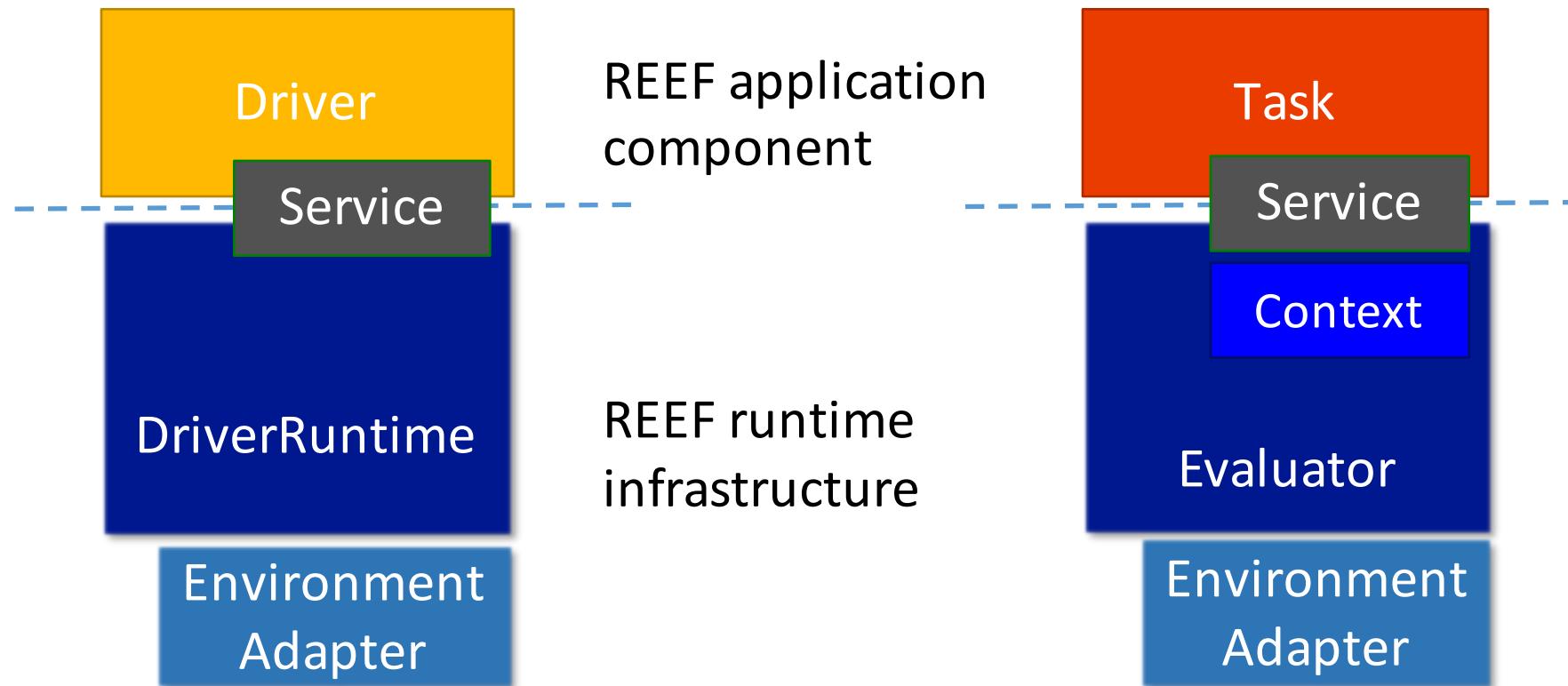


- ✓ Reusable control plane for coordinating data plane tasks
- ✓ Adaptation layer for resource managers
- ✓ Container and state reuse across tasks from heterogeneous frameworks
- ✓ Simple and safe configuration management
- ✓ Scalable local, remote event handling
- ✓ Java and C# (.NET) support

Architectural Choices Summary

1. Scalable master-workers control plane
2. Event processing framework: Wake
3. Configuration using dependency injection: Tang
4. Container and state reuse across heterogeneous frameworks

REEF Components



REEF Runtime Infrastructure

Driver Runtime

Hosts the application control-flow logic implemented in the Driver module

Environment Adapter

Translates Driver resource actions to the underlying resource manager protocol

REEF Runtime Infrastructure

Evaluator

A runtime environment on a container that can retain state within Contexts and execute Tasks [Task Runtime]

Context

A state management environment within an Evaluator, that is accessible to any Task hosted on that Evaluator

REEF Application Component

Driver

Application code that implements the resource allocation and task scheduling components.

1. Runtime Events
2. Evaluator Events
3. Task Events

Task

Application code executed in an Evaluator.

Task has access to its configuration parameters and the Evaluator state, which is exposed as Contexts

REEF Runtime Infrastructure and Application Component

DEVIEW
2015

Service

Factors out core functionalities that can be re-used across a broad range of applications; adds extensibility

REEF Services

Storage

Key-value store
DataLoader, OutputService

Network

Name-based messaging
Elastic group communication

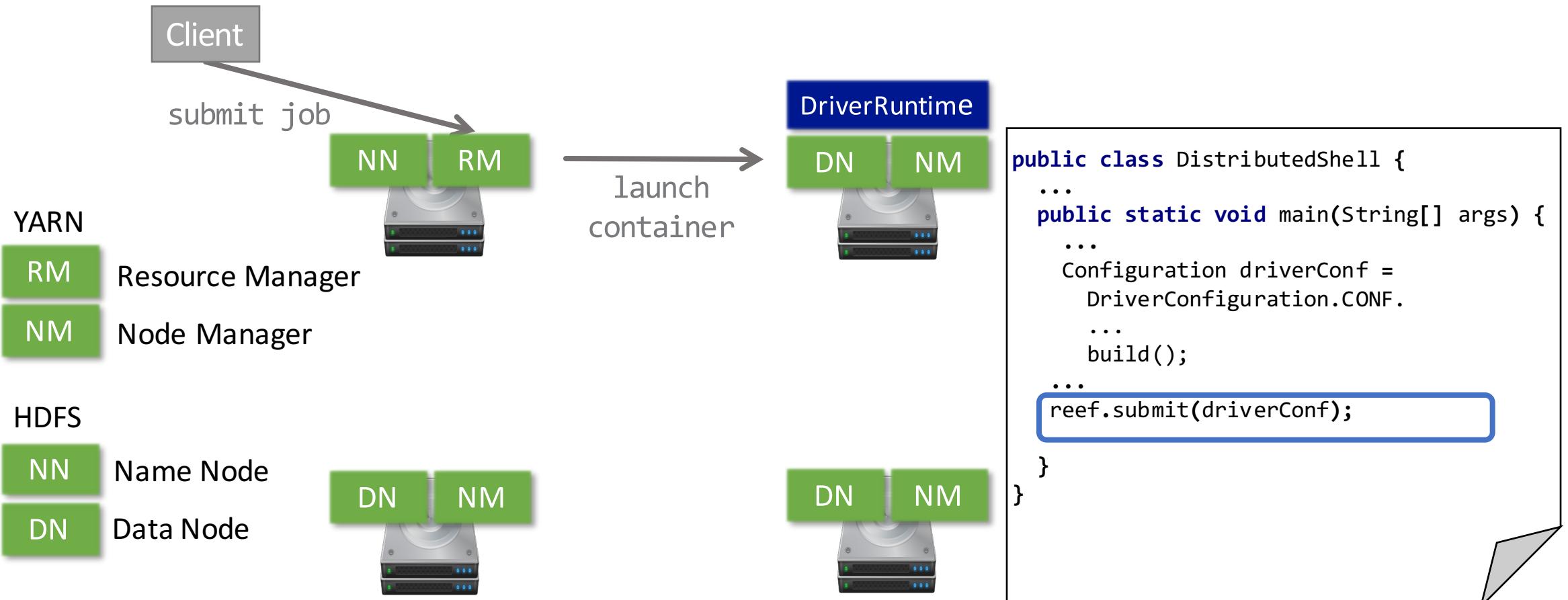
State Management

Checkpointing

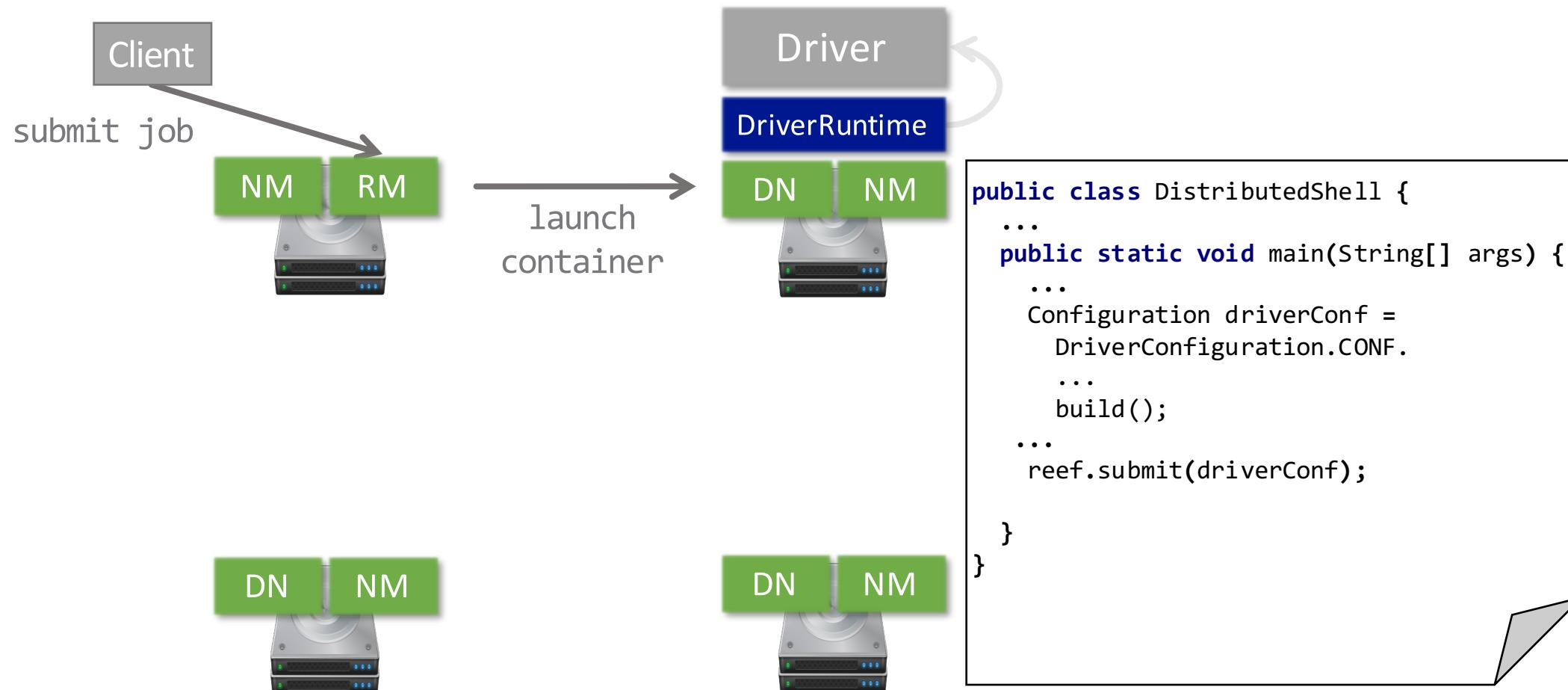
Monitoring

Profiling, HTTP server

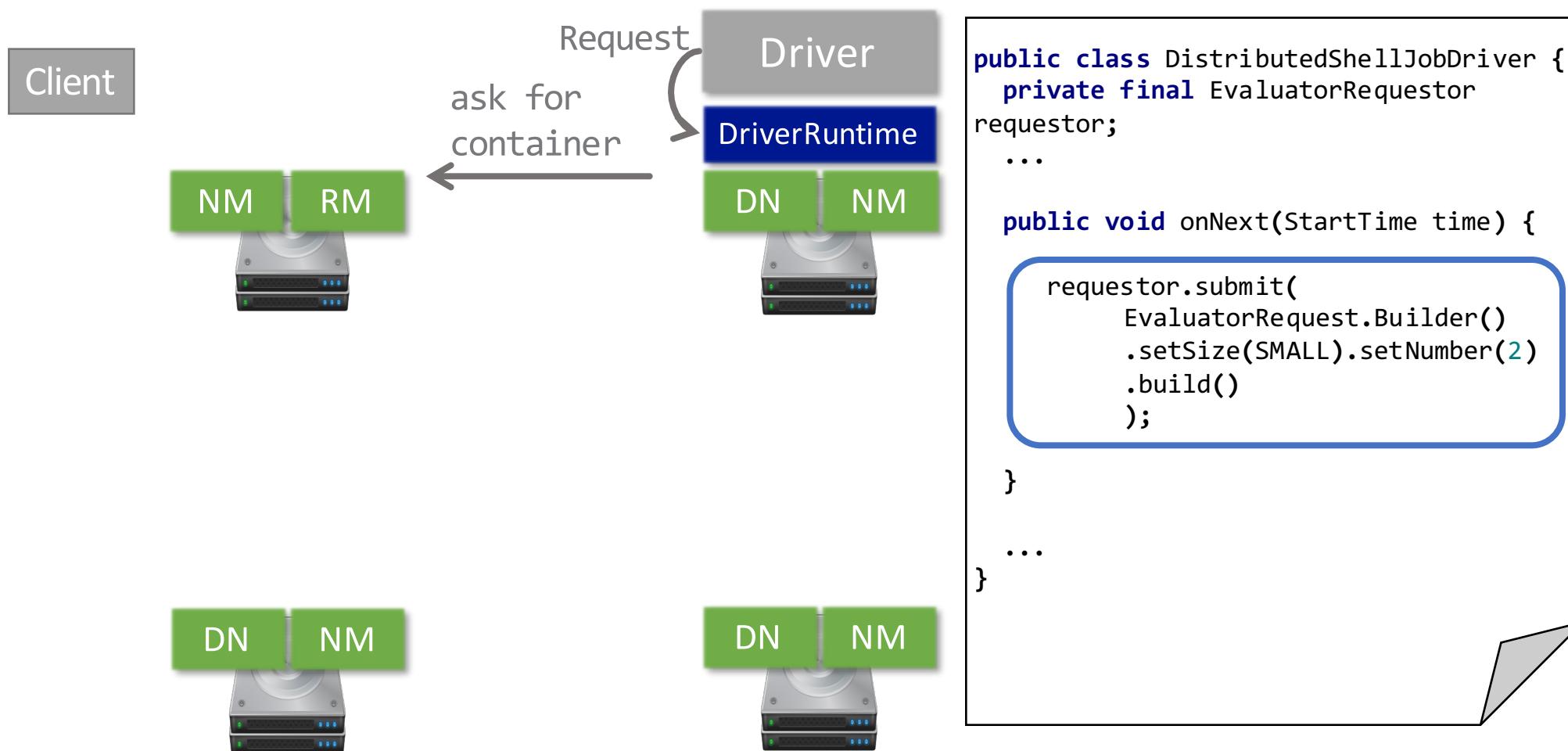
REEF Control Flow: Job Life Cycle



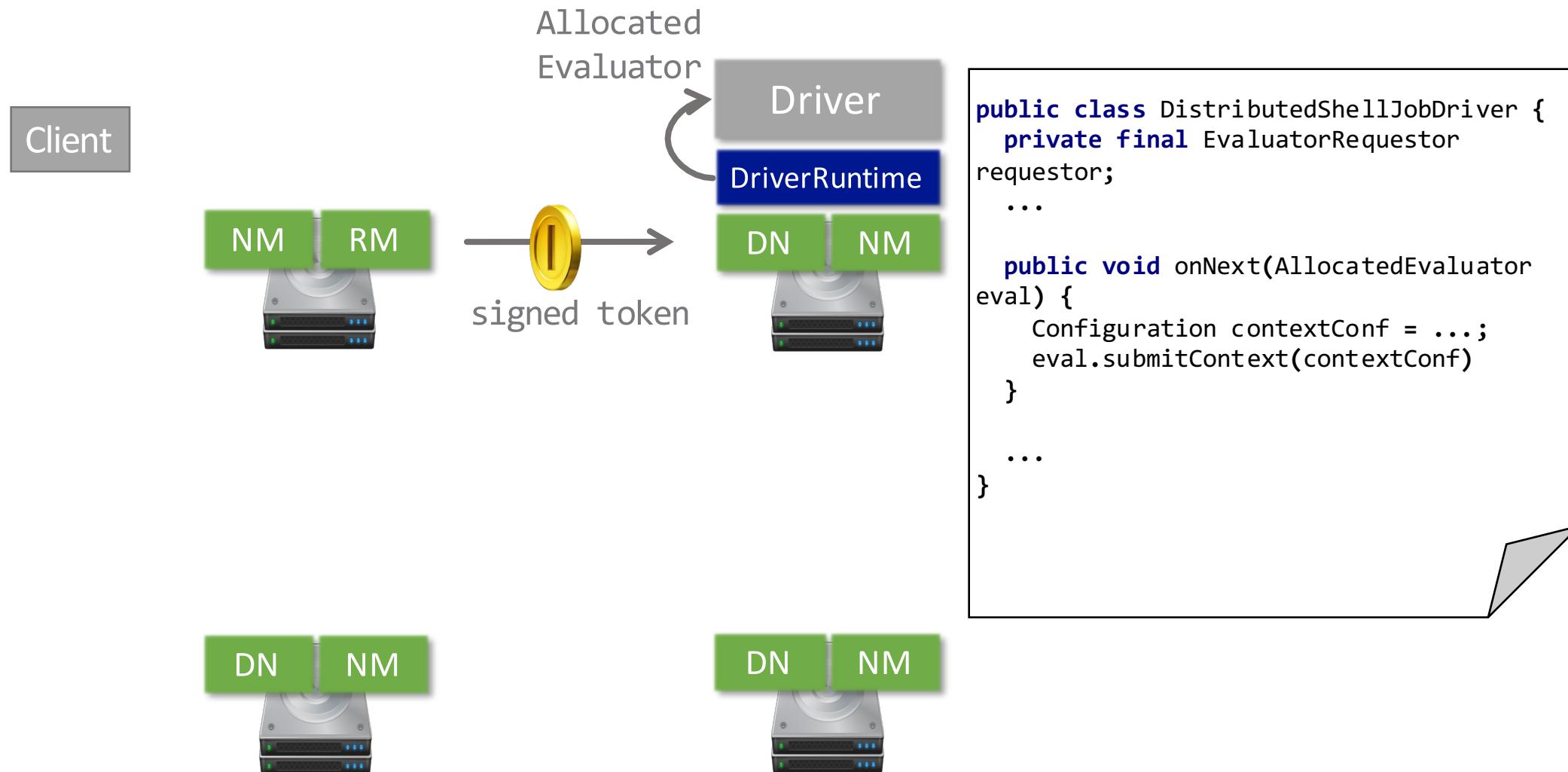
REEF Control Flow



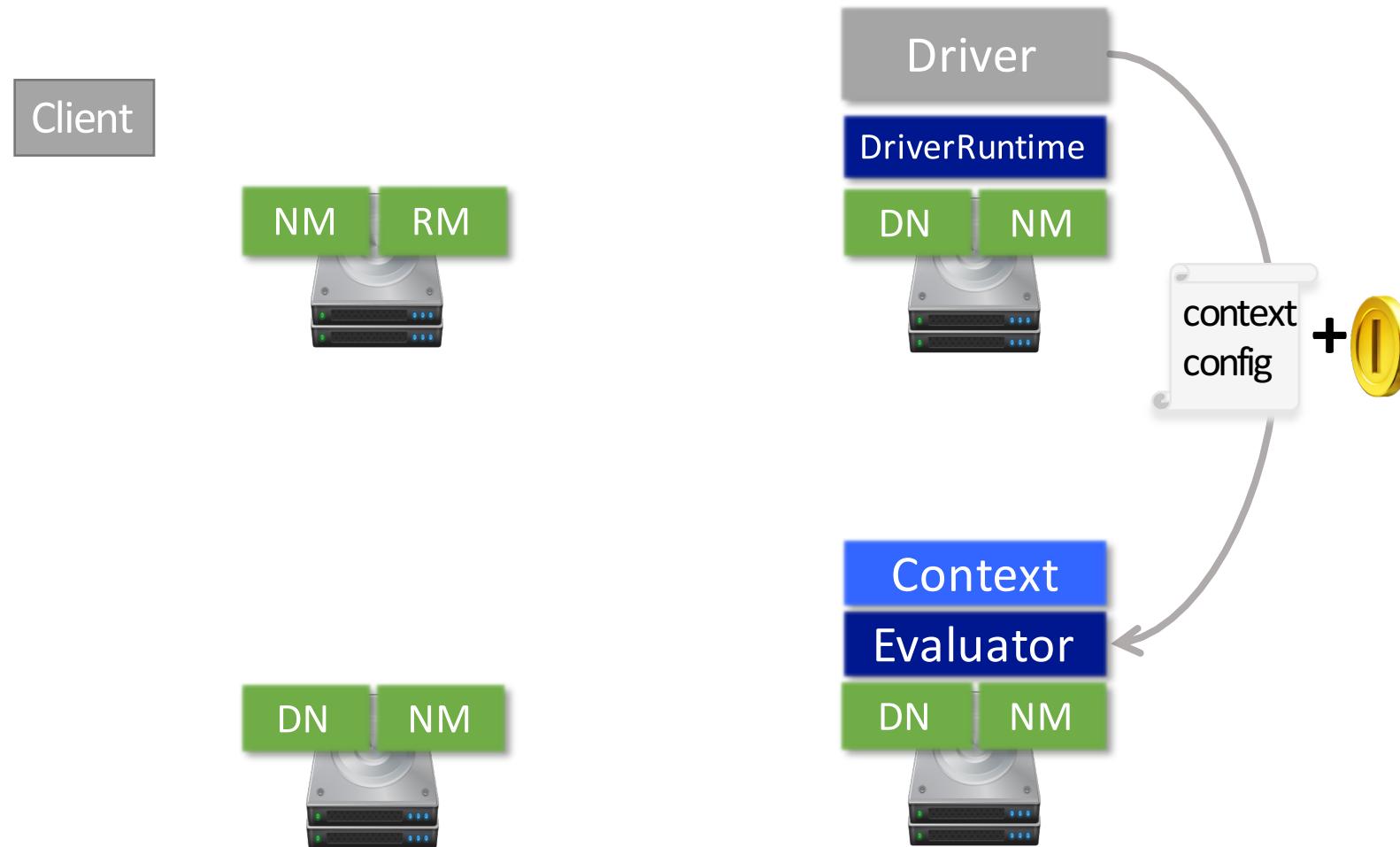
REEF Control Flow



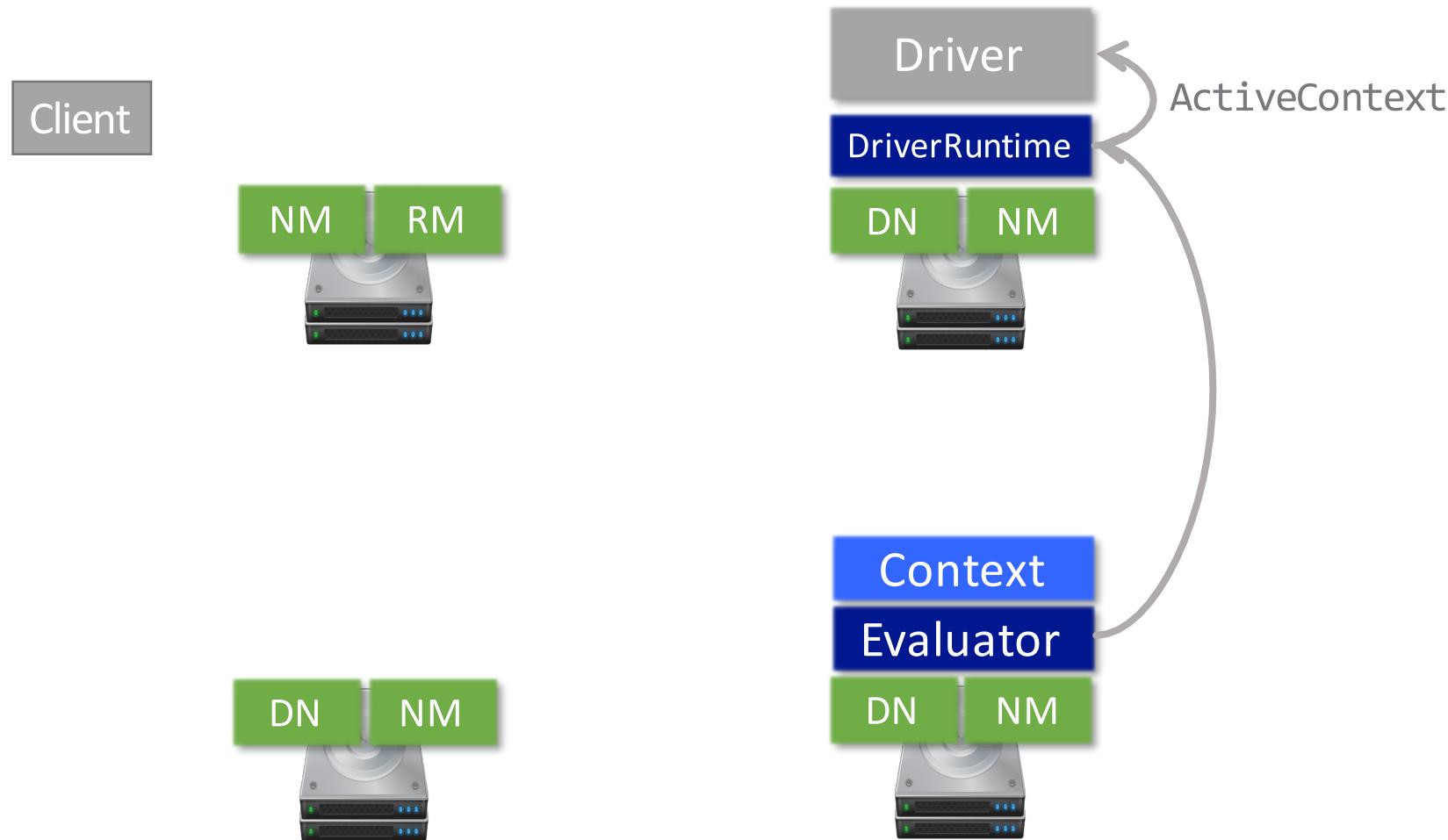
REEF Control Flow



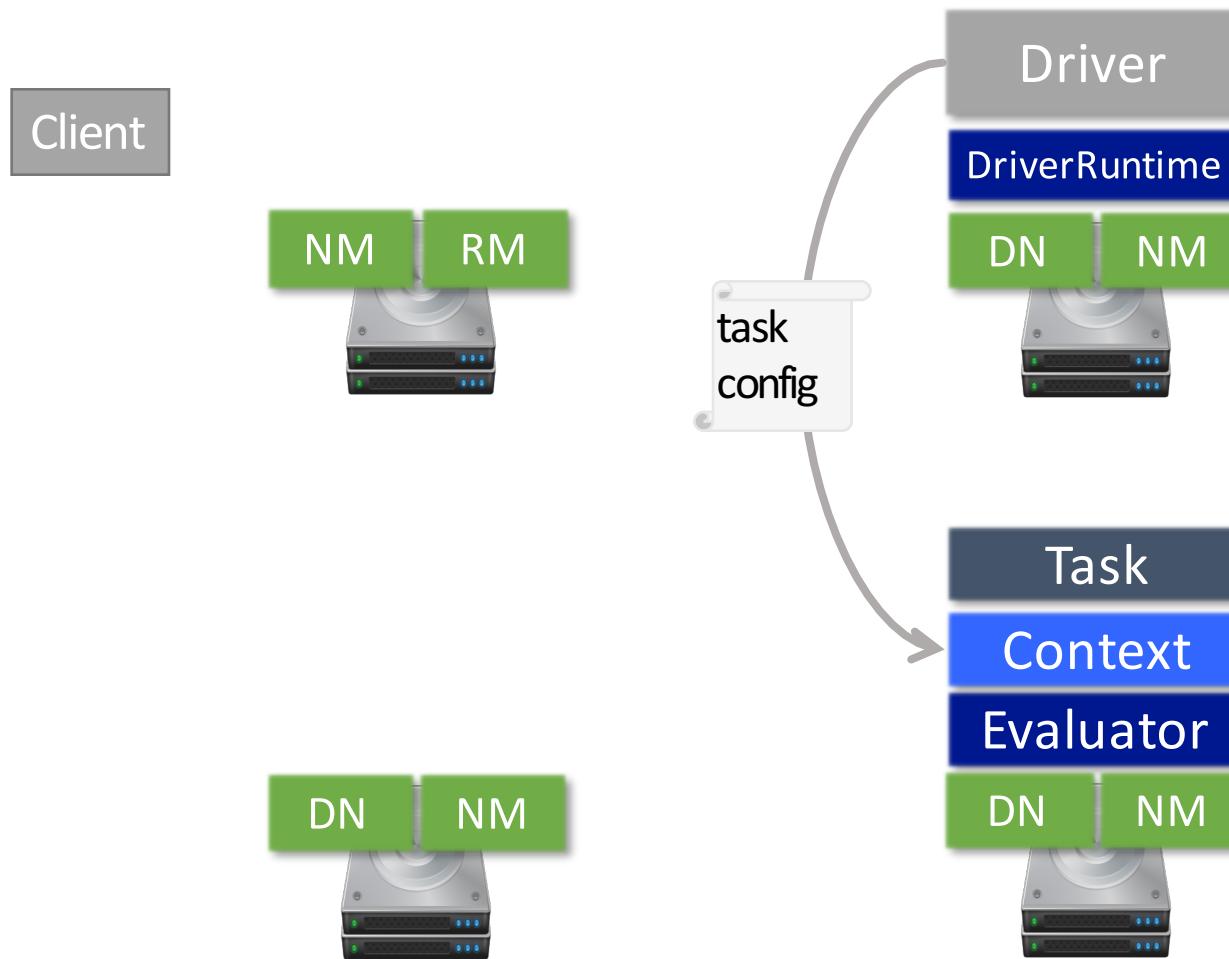
REEF Control Flow



REEF Control Flow



REEF Control Flow



```
public class DistributedShellJobDriver {  
    private final String cmd = "dir";  
  
    [...]  
  
    public void onNext(ActiveContext ctx) {  
        final String taskId = [...];  
  
        Configuration taskConf =  
            Task.CONF  
                .set(IDENTIFIER, "ShellTask")  
                .set(TASK, ShellTask.class)  
                .set(COMMAND, this.cmd)  
                .build();  
  
        ctx.submitTask(taskConf);  
    }  
  
    [...]  
}
```

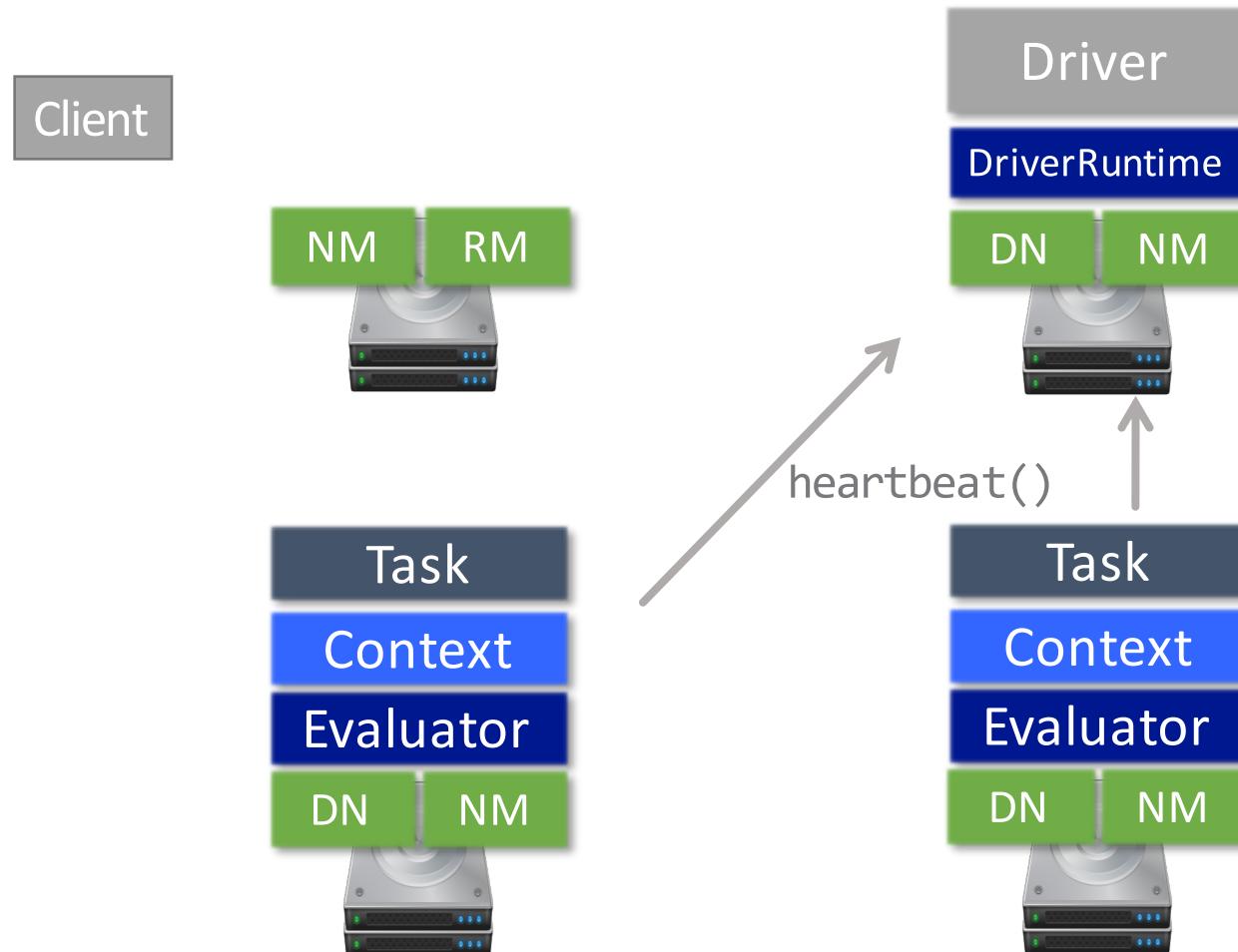
REEF Control Flow

Client

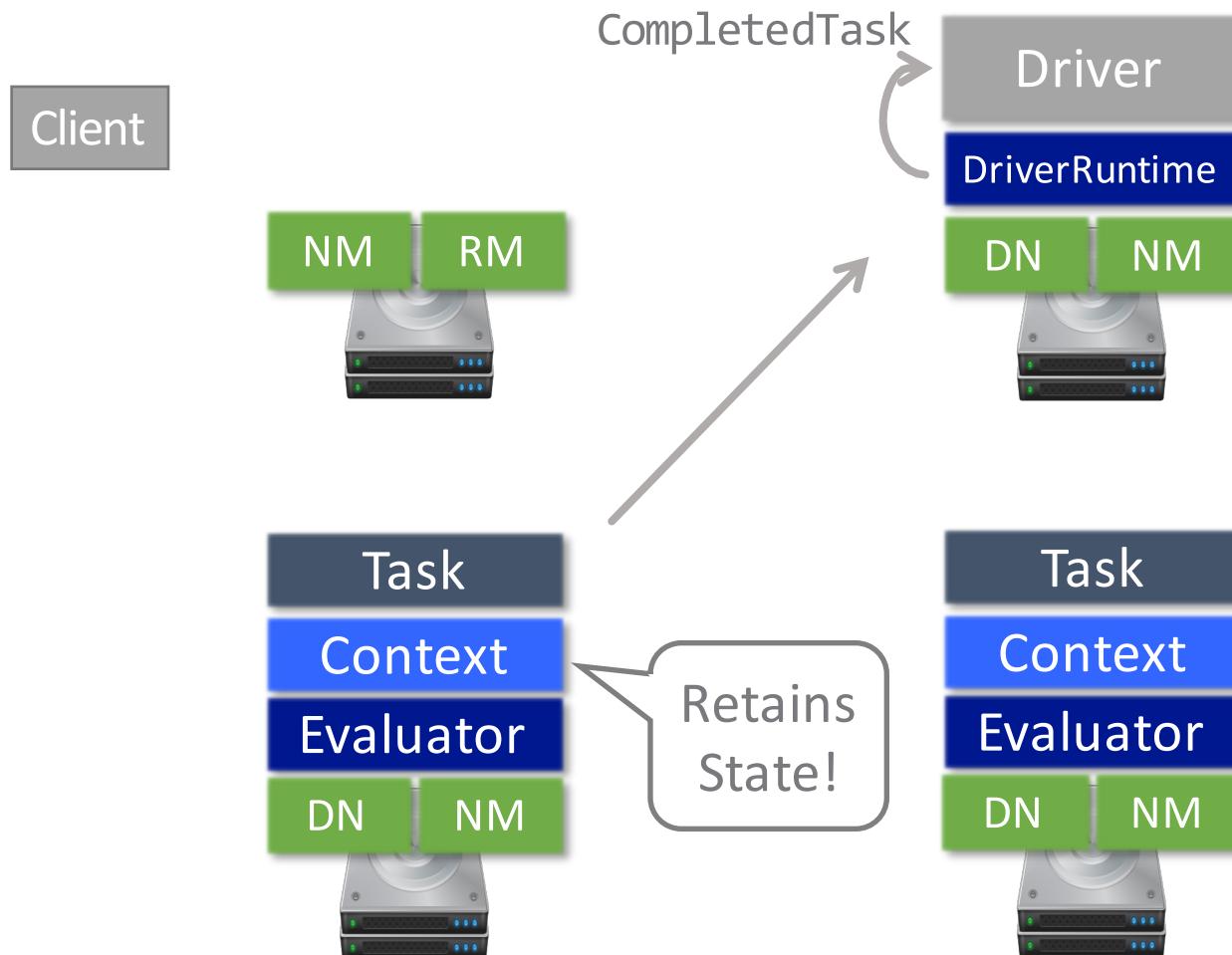


```
class ShellTask implements Task {  
  
    private final String command;  
  
    @Inject  
    ShellTask(@Parameter(Command.class)  
String c) {  
    this.command = c;  
}  
  
    private String exec(final String  
command){  
    ...  
}  
  
    @Override  
    public byte[] call(byte[] memento) {  
        String s = exec(this.cmd);  
        return s.getBytes();  
}  
}
```

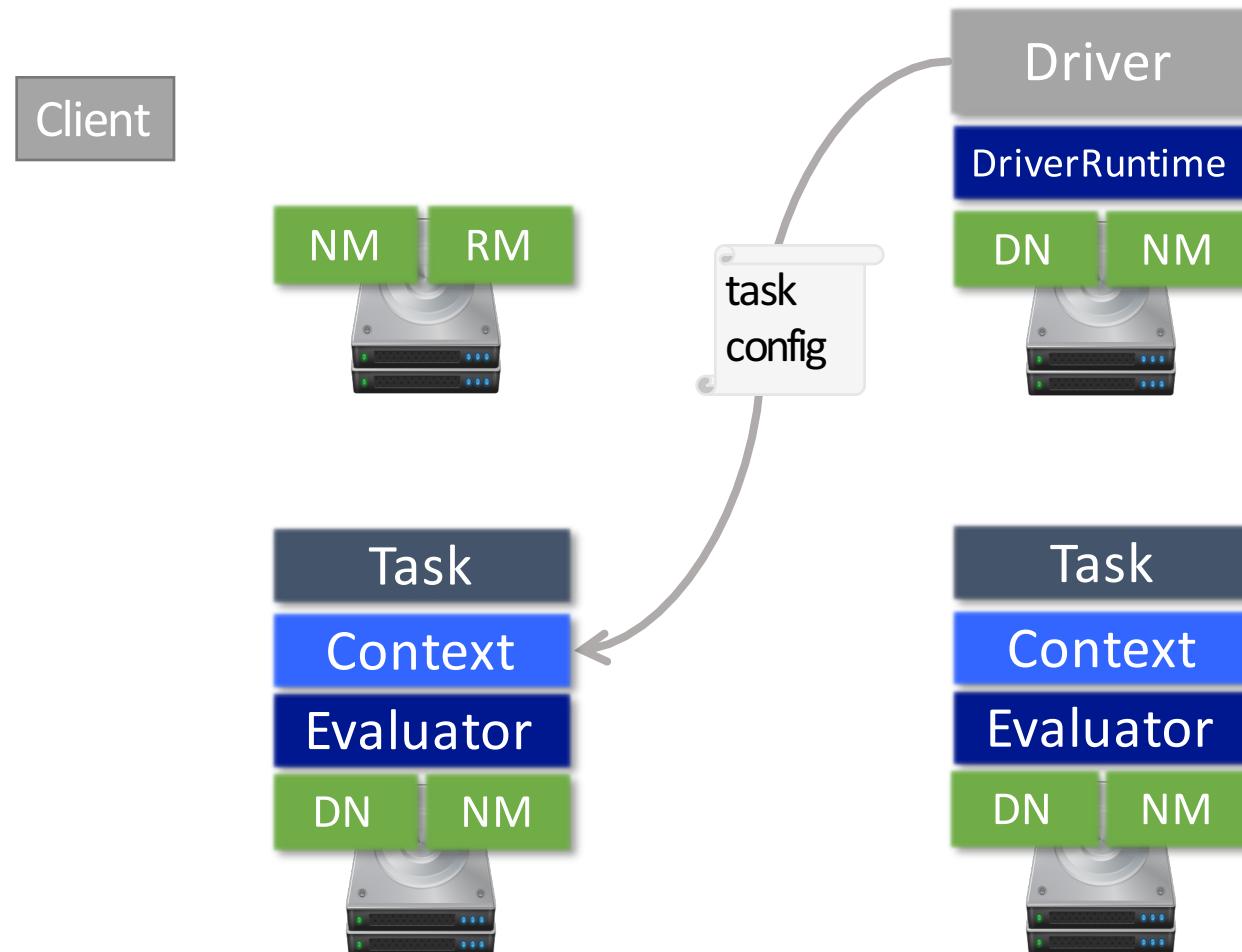
REEF Control Flow



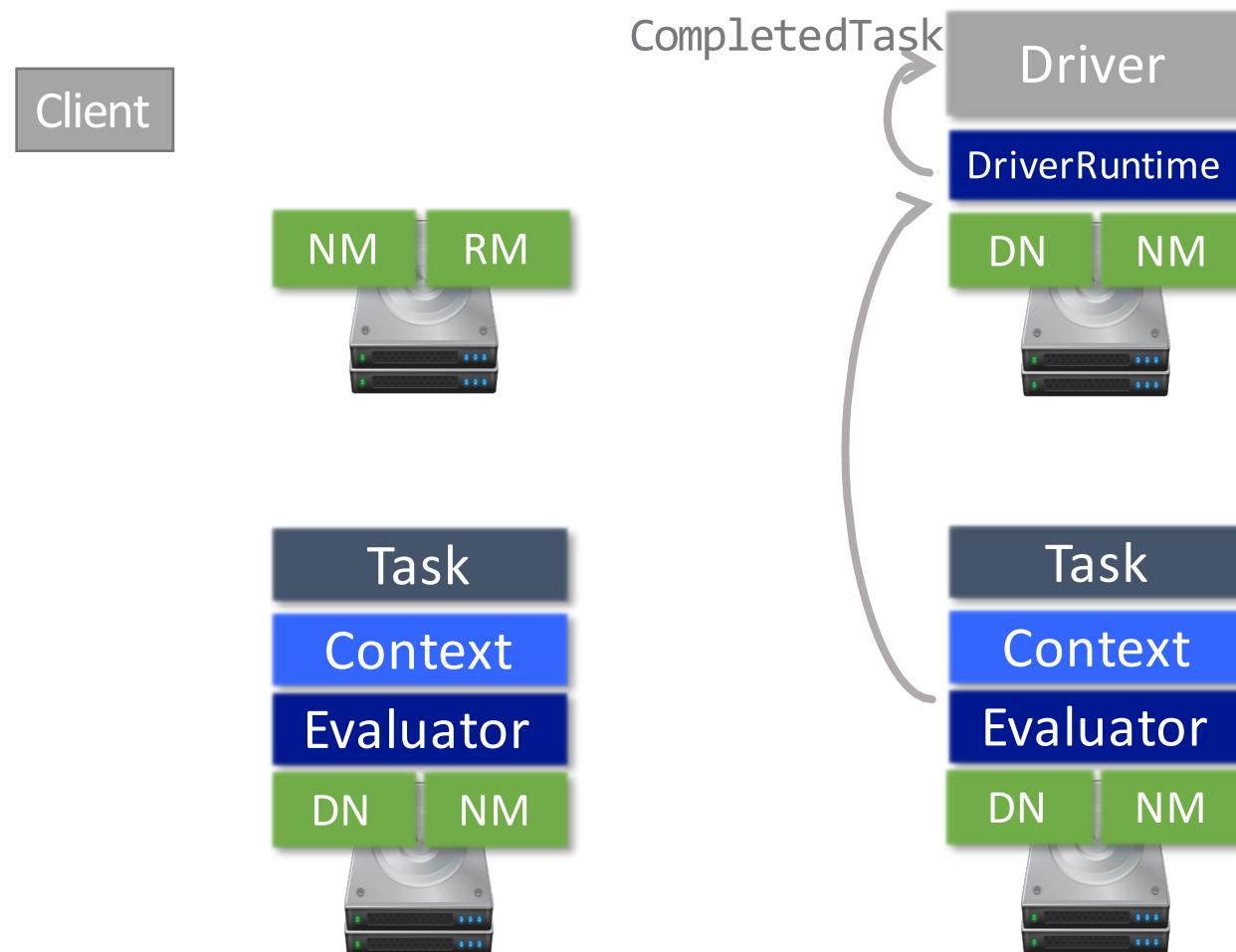
REEF Control Flow



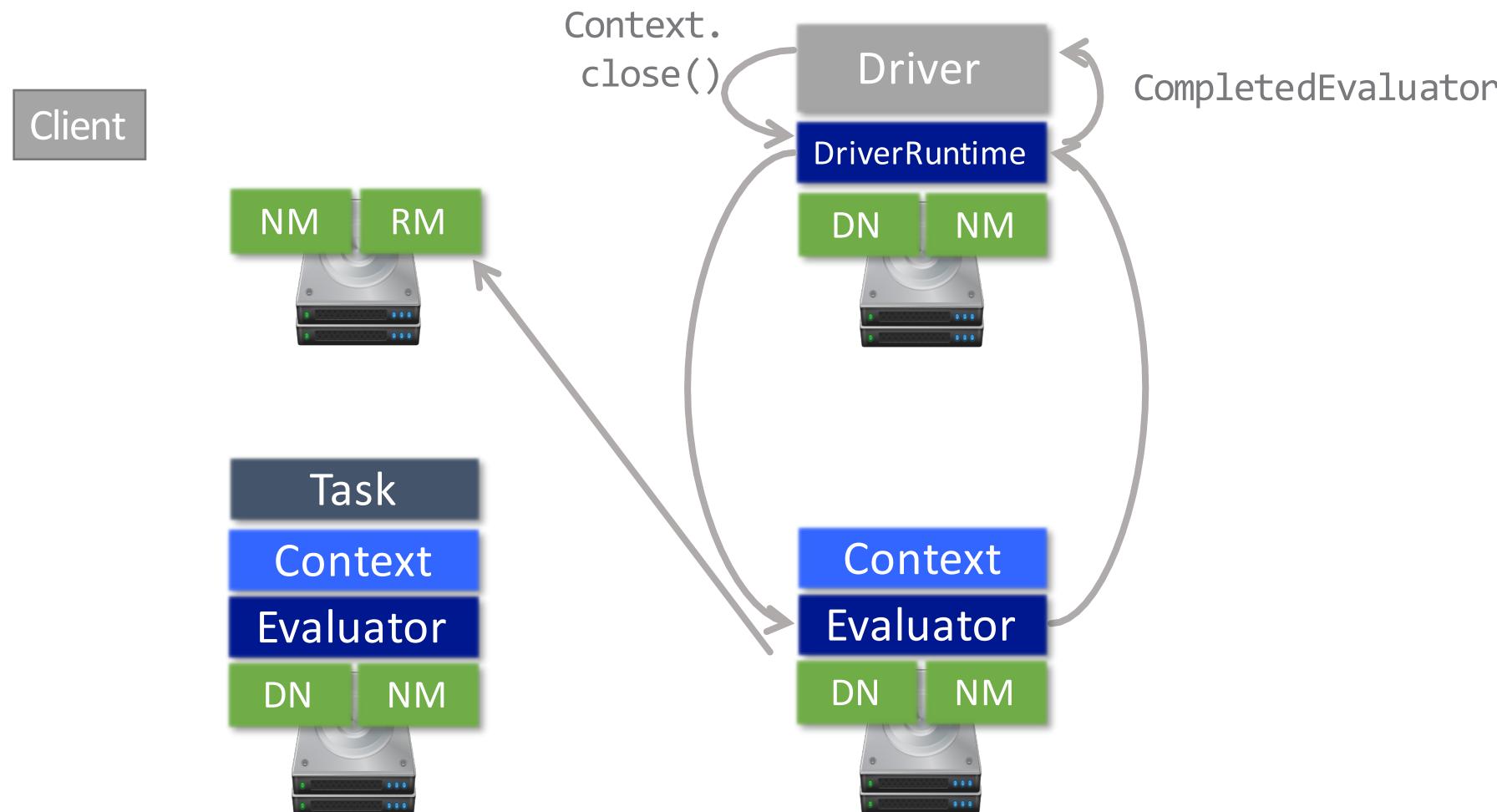
REEF Control Flow



REEF Control Flow



REEF Control Flow



Supported Execution Environments

- Local Machine: testing
- YARN: Hadoop resource manager (containers)
- Mesos: resource manager started at UC Berkeley (resource offers)
- HDInsight: Azure Hadoop offering

Wake

- Push-based event handlers, observers
 - User code: Wake event handlers that react to various REEF events
- Stage to execute event handlers: synchronous or asynchronous execution
- Thread management
- Scalable remote messaging

Tang

- Configuration using dependency injection that restricts dynamic bind patterns
- Tang specification: binding patterns
 - Bind an implementation to an interface
 - Bind a value to a configuration parameter
- Tang configuration language semantics
 - “Configuration” are just data: cross-language support
 - Configuration options can be set at most once: Immutability
 - A large subset of Tang’s public API is commutative

REEF Implementation

Lines of code by component and language (September 11, 2015)

| | C# | Java | CPP | Total |
|----------|--------|--------|-------|--------|
| Tang | 8,951 | 7,454 | 0 | 16,405 |
| Wake | 4,677 | 5,268 | 0 | 9,945 |
| REEF | 13,821 | 24,281 | 1,957 | 40,059 |
| Services | 4,527 | 11,620 | 0 | 16,147 |
| Total | 31,976 | 48,623 | 1,957 | 82,556 |

Applications can mix and match Driver side event handlers in Java and C# with any number of Java and C# Evaluators

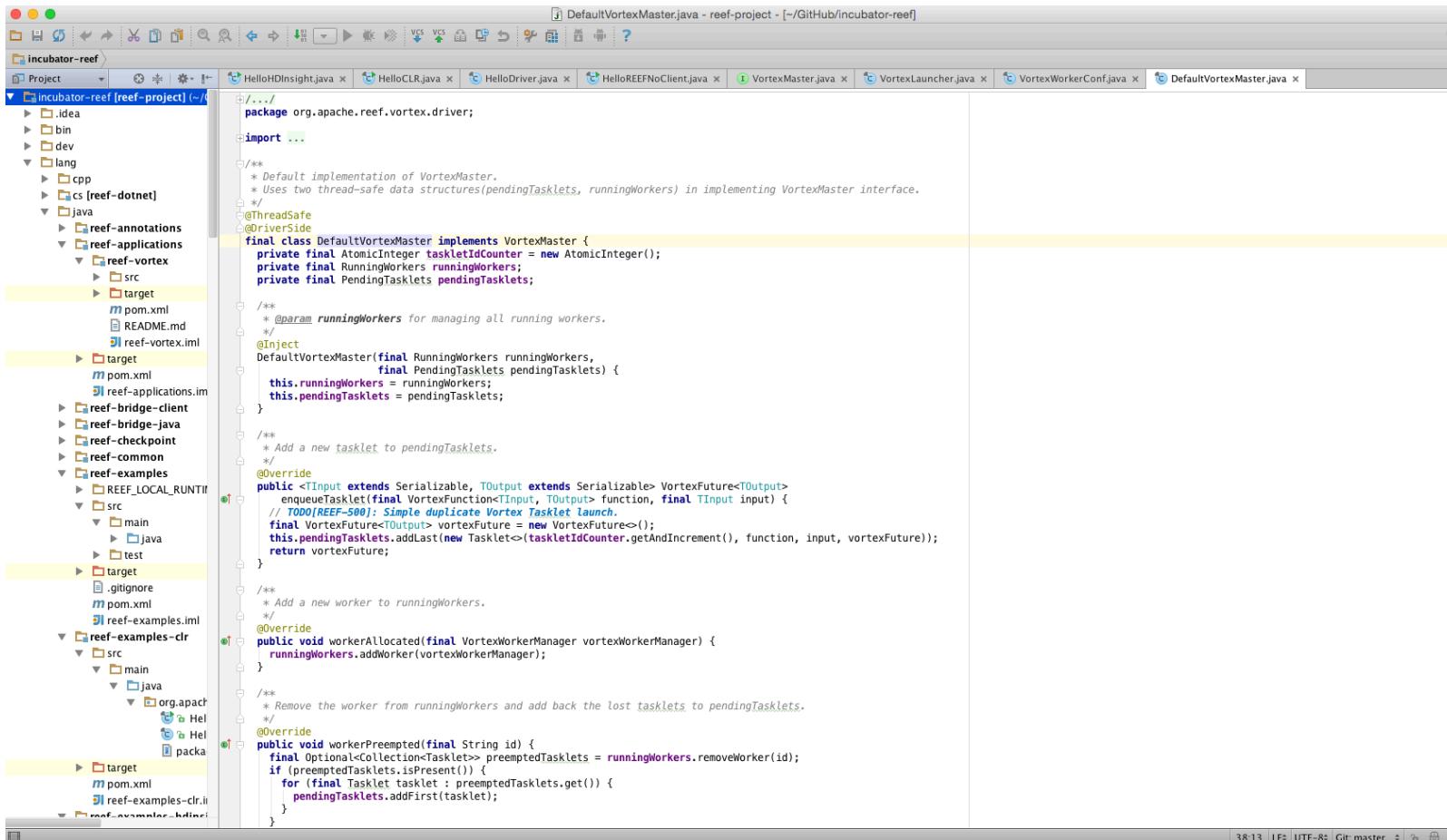
A distributed control flow framework that provides integration across language boundaries

Get Started with REEF

REEF Project Pointers

- REEF web site: <http://reef.incubator.apache.org>
- Github mirrored site: <https://github.com/apache/incubator-reef>
- Some basic examples to start with
 - HelloREEF*: HelloREEF, HelloREEFYarn, HelloREEFMesos, ...
 - TaskScheduler: job scheduler to schedule embarrassingly parallel jobs and scale out and in resources to handle fluctuating demands
- Tutorials: <https://cwiki.apache.org/confluence/display/REEF/Tutorials>

Work with Your Favorite IDE

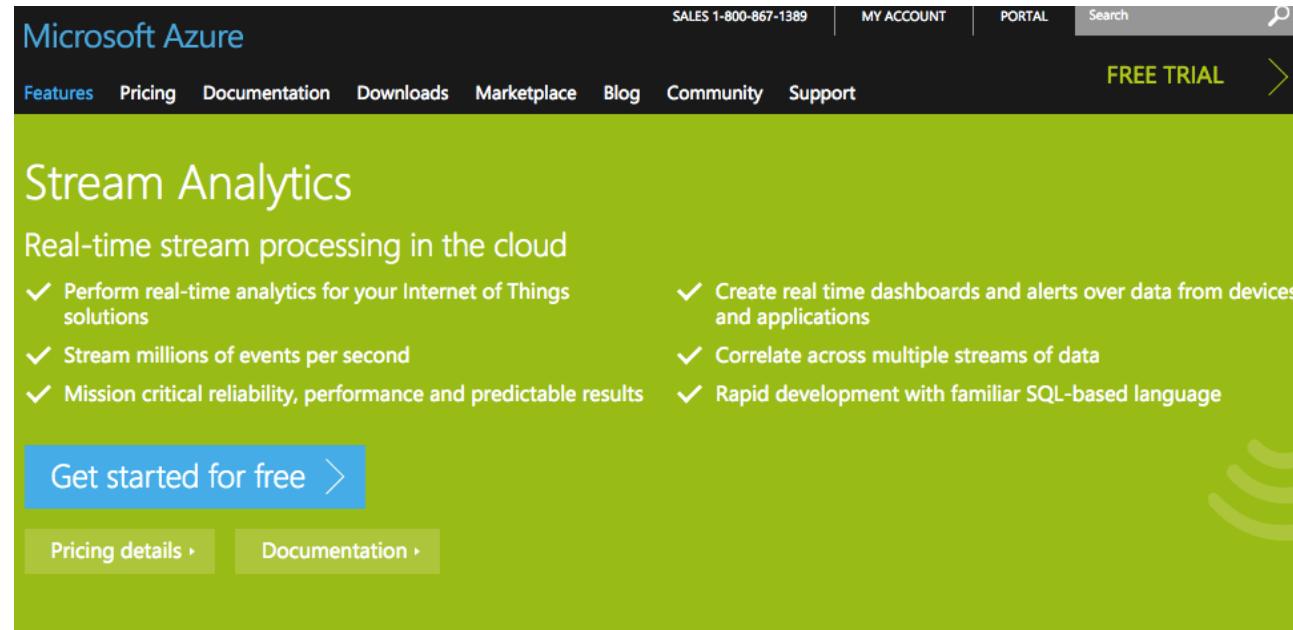


The screenshot shows an IDE interface with the following details:

- Title Bar:** DefaultVortexMaster.java - reef-project - [~/GitHub/incubator-reef]
- Toolbar:** Standard IDE toolbar with icons for file operations, search, and navigation.
- Project Explorer:** Shows the `incubator-reef` project structure. The `reef-project` module is expanded, displaying sub-directories like `.idea`, `bin`, `dev`, `lang`, `java`, `reef-annotations`, `reef-applications`, `reef-vortex`, `target`, `reef-bridge-client`, `reef-bridge-java`, `reef-checkpoint`, `reef-common`, `reef-examples`, and `reef-examples-clr`. Each module has its own `pom.xml` and `README.md` files.
- Code Editor:** The main window displays the `DefaultVortexMaster.java` file. The code implements the `VortexMaster` interface, managing tasklets and workers. It includes annotations like `@ThreadSafe`, `@DriverSide`, and `@Inject`. The code uses Java 8 features such as `AtomicInteger`, `Optional`, and `Stream`.
- Status Bar:** Shows the current time (38:13), file encoding (UTF-8), and Git status (master).

Why Would You Want REEF?

Microsoft Azure Services on REEF



The screenshot shows the Microsoft Azure Stream Analytics landing page. At the top, there's a navigation bar with links for Sales (1-800-867-1389), My Account, Portal, Search, and a Free Trial button. Below the navigation is a main heading 'Stream Analytics' and a sub-heading 'Real-time stream processing in the cloud'. To the left, there's a list of features: 'Perform real-time analytics for your Internet of Things solutions', 'Stream millions of events per second', and 'Mission critical reliability, performance and predictable results'. To the right, another list includes: 'Create real time dashboards and alerts over data from devices and applications', 'Correlate across multiple streams of data', and 'Rapid development with familiar SQL-based language'. At the bottom, there's a large blue 'Get started for free >' button, and two smaller buttons for 'Pricing details' and 'Documentation'.

Uncover Real-time Insights

Learn how Microsoft's stream analytics service in the cloud enables you to rapidly develop and deploy a low cost real-time analytics solution to uncover real-time insights from devices, sensors, infrastructure, and applications. It will enable various opportunities including Internet of Things (IoT) scenarios such as real-time remote management and monitoring or gaining insights from devices like mobile phones or connected cars.



Azure Stream Analytics (ASA):
a fully managed stream processing service
offered in the Microsoft Azure Cloud

Azure Stream Analytics on REEF

- REEF Driver
 - Compiles and optimizes (taking user budget into consideration) a query into a data-flow of processing stages
 - Uses the stage data-flow to formulate a request for resources: an Evaluator per stage instance
 - Driver restartability
- REEF Task
 - Executes the stage instance work on an assigned partition
- REEF Services used
 - Communication service built on Wake that allows sending messages with logical identifiers
 - Checkpointing service

VMWare CORFU on REEF

- CORFU (NSDI 2012): a distributed logging tool providing applications with consistency and transactional services at extremely high throughput
- CORFU master deployed in a Driver
 - A logging unit fails, the Driver is informed, and the CORFU master reacts
- REEF Service for triggering checkpointing and restarting a Driver from a checkpoint
- REEF decouples CORFU's resource deployment from its state, allowing CORFU to be elastic

Why Would You Want REEF?

- New data processing framework
 - Cloud Data Runtime (CDR)
 - Iterative Map-Reduce-Update (IMRU)
 - Elastic ML in CAY
- New system services
 - Elastic “memory”, “communication”, and “cache” in CAY
 - Vortex: harnessing transient cluster resources in CAY

How Does REEF Make Life Easier?

- REEF provides high-level component abstraction:
Driver, Task, and Service
- REEF application developers simply write handlers that react to various events
- REEF runtime handles common data processing control-plane functionalities

What's Next?

- We're ready to graduate from the Incubator
 - Expected to be the 3rd Apache TLP project initiated from Korea (after Hama, Tajo)
- More production adoption
- New REEF primitives
- More systems built on top of REEF

From REEF To CAY



CAY: A Unified Big Data Analytics Stack

- 미래창조과학부 sw스타랩 프로젝트:
(SW 스타랩) 빅데이터 플랫폼 연구실
- Unification: development, operation
 - Big Data operating system layer
 - Unified data management
 - Unified programming
- New/enhanced data processing engines

Elastic Memory

- Elastically change memory resources for in-memory big data analytics
- Solves the problem of the static allocation of resources for low-latency interactive services, machine learning, etc.

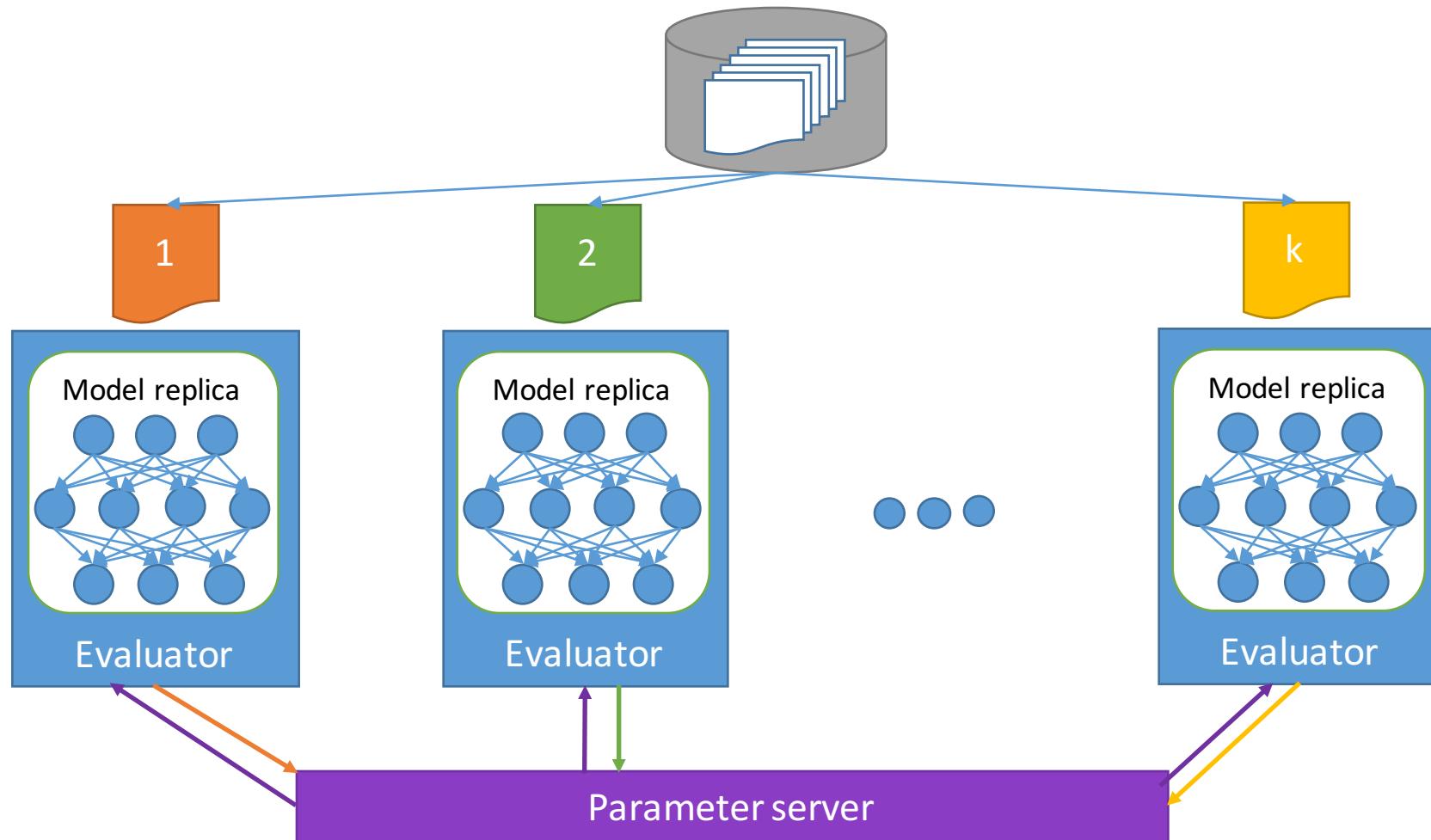
Elastic Communication

- Elastically configure communication among nodes
- Elastic shuffle
- Elastic group communication
 - Scatter, gather, broadcast, reduce, ...

Elastic Machine Learning (SNU, Microsoft)

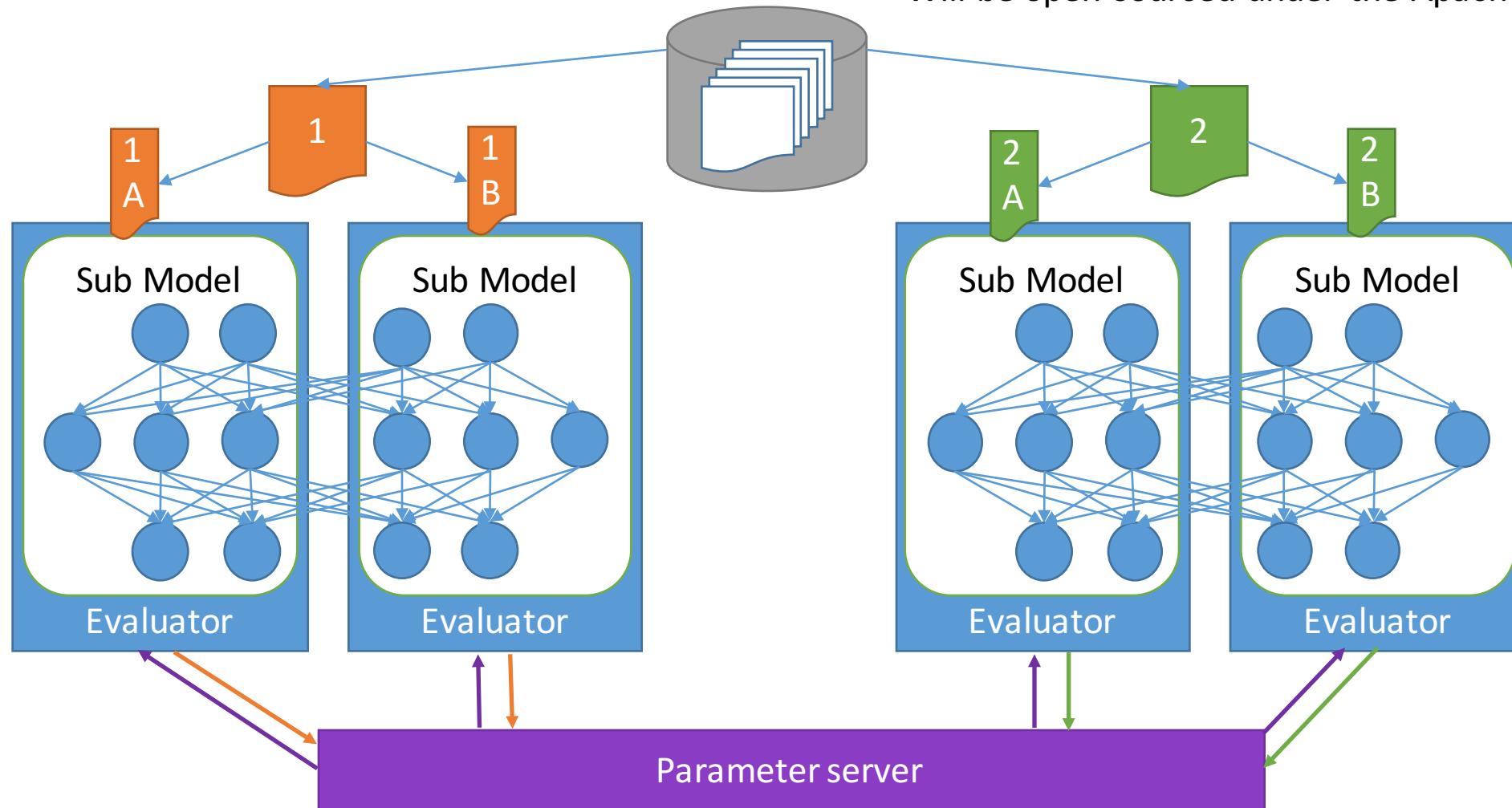
- Goal: optimize large-scale ML execution automatically
- Built on “elastic” system services
- Elastic ML runtime & optimizer
- A new breed of elastic Machine Learning algorithms

Large-scale Deep Learning (SNU, SKT)



Large-scale Deep Learning (SNU, SKT)

* Will be open sourced under the Apache License 2



Stream processing

- 초대용량 마이크로 스트림 질의 처리 플랫폼
(삼성미래기술육성사업)
- Applications
 - Personalized personal digital assistant
 - Real-time disease control
 - Smart home, smart building

Vortex (SNU, Microsoft Research)

- Harness volatile resources available in the datacenter
 - Large idle resources in production systems
- Massively parallel tasklets
- Rearchitect distributed analytics execution (e.g., data flow graph execution) to perform well in such volatile environments

References

- Web links
 - <http://reef.incubator.apache.org> (REEF project site)
 - <http://cmslab.snu.ac.kr> (My research lab site)
- Publication
 - REEF: Retainable Evaluator Execution Framework, ACM SIGMOD 2015.
 - Making Sense of Performance in Data Analytics Frameworks, USENIX NSDI 2015
 - Elastic Memory: Bring Elasticity Back To In-Memory Big Data Analytics, HotOS 2015.

CMS Lab <http://cmslab.snu.ac.kr>

We are hiring developers and
postdoc researchers!

Contact: Byung-Gon Chun (bgchun@snu.ac.kr)
Seoul National University



REEF

DEVIEW
2015

딥러닝에서 스트림처리까지: 빅데이터 분석 메타 프레임워크 Apache REEF

2015년 9월 14일

전병곤
서울대학교 컴퓨터공학부
bgchun@snu.ac.kr