GATK Best Practices for Variant Discovery

Pipelining with WDL and Cromwell

Overview/recap

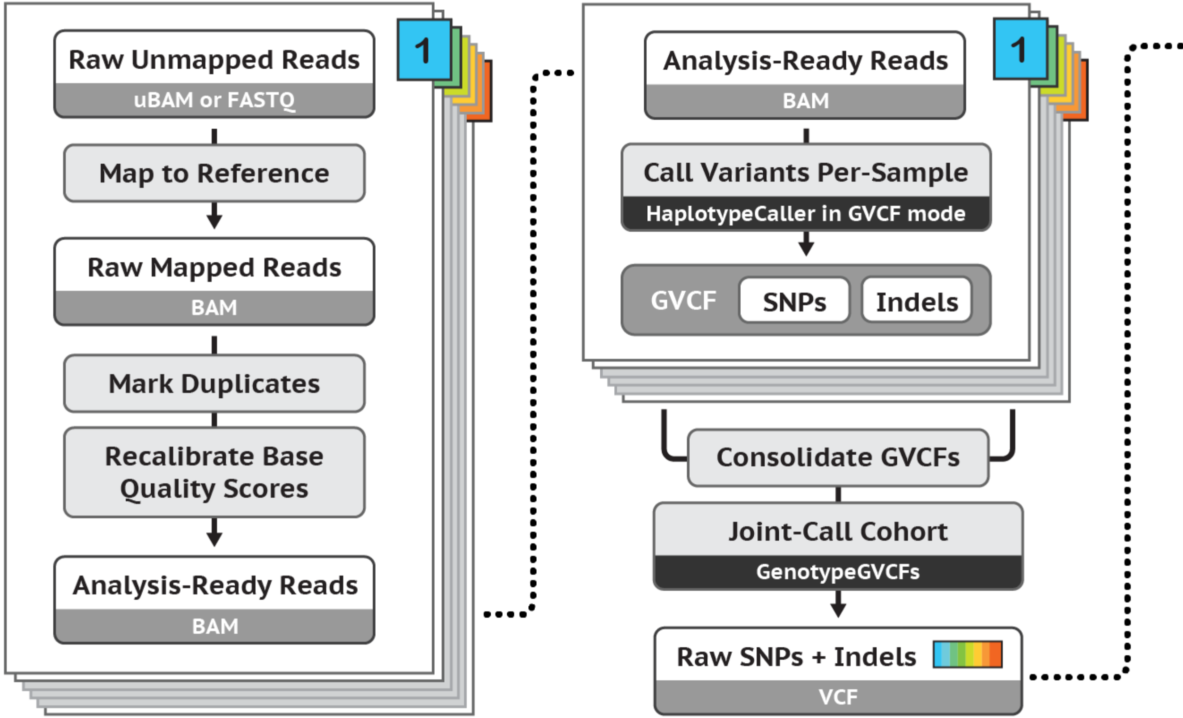
BROAD INSTITUTE

http://software.broadinstitute.org/gatk/

gatk

Goal: automate processing by running scripted pipelines

- Automate repetitive tasks
- Increase auditability and reproducibility
- Reduce human error
- Reduce time spent re-implementing the wheel



WDL = Workflow Definition Language

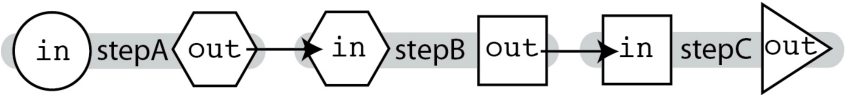
```
workflow myWorkflowName {
    File my_ref
    File my_input
    String name
  call task_A {
    input: ref= my_ref, in= my_input, id= name
  }
  call task_B {
    input: ref= my_ref, in= task_A.out
  }
}
task task_A {      ...      }
task task_B {      ...      }
```

```
task task_A {
    File ref
    File in
    String id
  command {
    do_stuff -R ${ref} -I ${in} -O ${id}.ext
  }
  runtime {
    docker: “my_project/do_stuff:1.2.0”
  }
  output {
    File out= “${id}.ext”
  }
}
```

http://www.openwdl.org

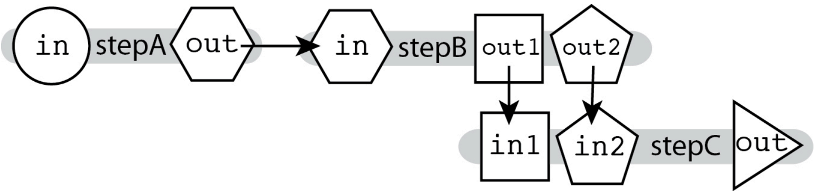
Basic plumbing options in WDL

LINEAR CHAINING



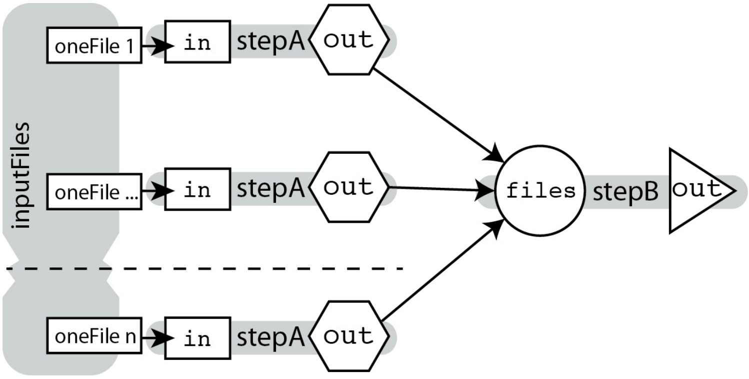
```
call stepA
call stepB { input: in=stepA.out }
call stepC { input: in=stepB.out }
```

MULTI-IN/OUT



```
call stepC { input :
                in1=stepB.out1,
                in2=stepB.out2 }
```

SCATTER-GATHER

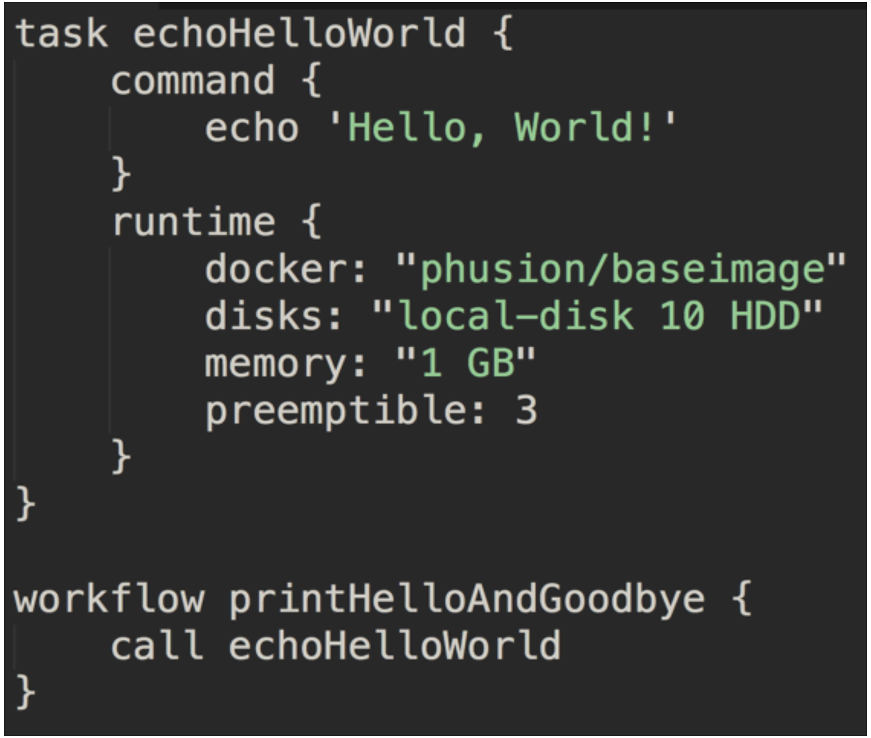


```
Array[File] inputFiles

scatter(oneFile in inputFiles) {
    call stepA { input: in=oneFile }
}

call stepB { input: files=stepA.out }
```

WDL runtime parameters

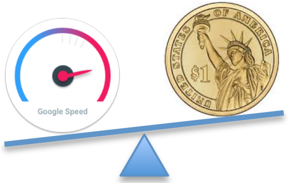


```
task echoHelloWorld {
    command {
        echo 'Hello, World!'
    }
    runtime {
        docker: "phusion/baseimage"
        disks: "local-disk 10 HDD"
        memory: "1 GB"
        preemptible: 3
    }
}

workflow printHelloAndGoodbye {
    call echoHelloWorld
}
```

containers

resourcing

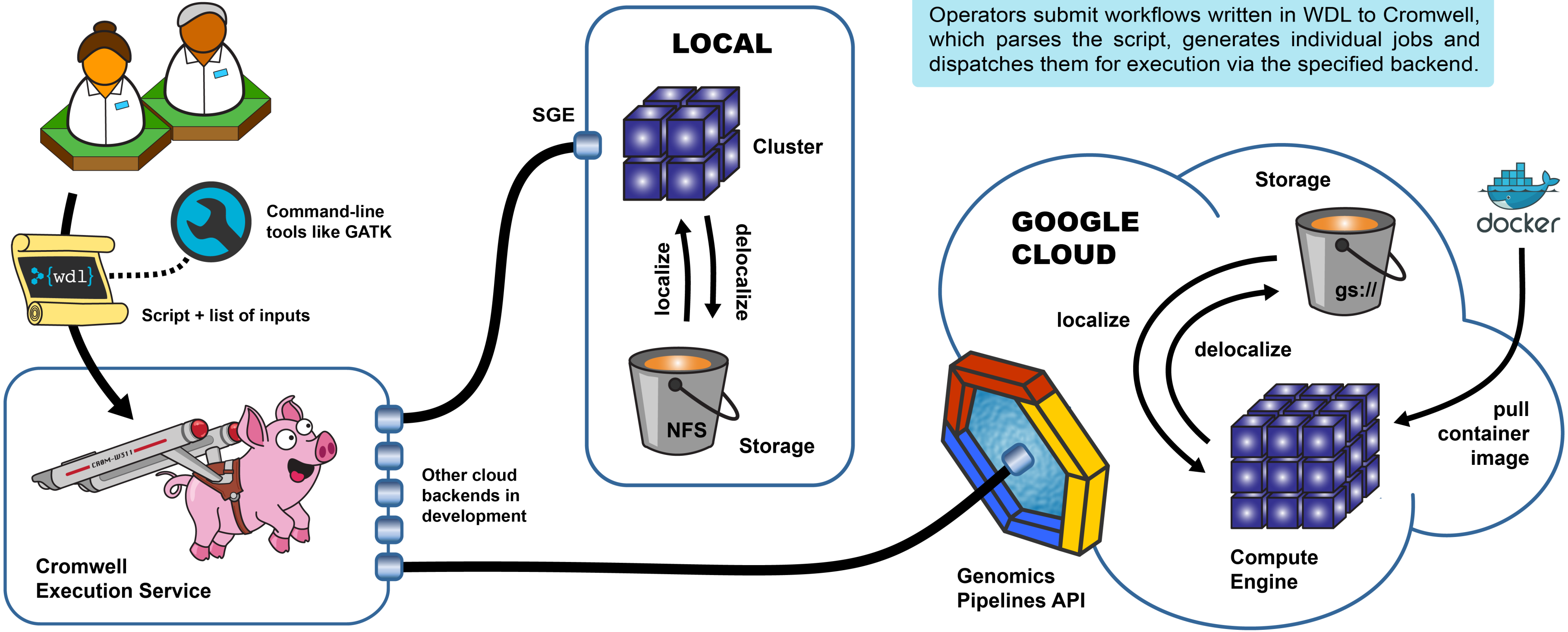


cost savings!

https://cloud.google.com/compute/docs/instances/preemptible

Write WDL -> Cromwell interprets -> runs jobs

Operators submit workflows written in WDL to Cromwell, which parses the script, generates individual jobs and dispatches them for execution via the specified backend.



Two main ways to run Cromwell

One-off

- Simple self-contained command

```
java -jar cromwell.jar \
     run hello.wdl \
     --inputs hello_inputs.json
```

- Appropriate for independent analysts

Server mode

- API endpoints
- More scalable
- Some devops needs
- Appropriate for production environments
- Call-caching!
  (aka "ka-ching")

Steps to running a WDL in standalone mode

- Validate syntax

```
java -jar womtool.jar validate hello.wdl
```

- Generate inputs JSON

```
java -jar womtool.jar inputs hello.wdl > hello_inputs.json
```

- Run

```
java -jar cromwell.jar run hello.wdl --inputs hello_inputs.json
```