Enabling Economical Genome Analyses through Optimization and Scalable Workflows

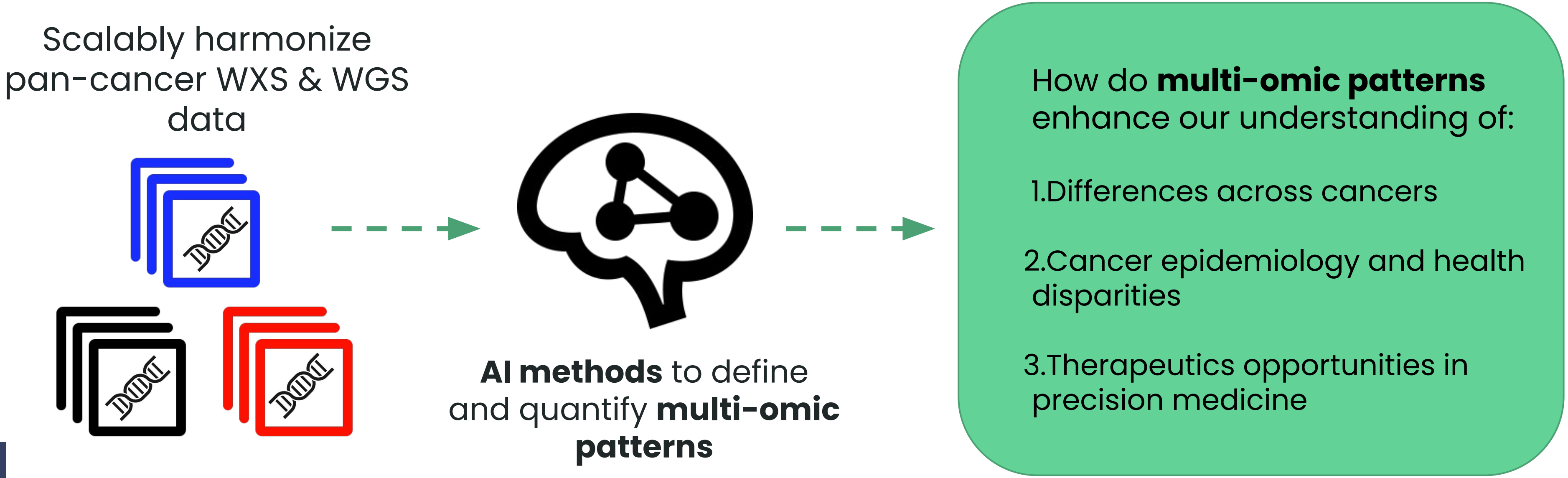
ParslFest 2024

Akila Perera, Jason J. Pitt

Cancer Science Institute of Singapore

Pitt Lab

Petabyte-scale Genomics to Fuel Precision Oncology



Scalably harmonize pan-cancer WXS & WGS data

AI methods to define and quantify **multi-omic patterns**

How do **multi-omic patterns** enhance our understanding of:

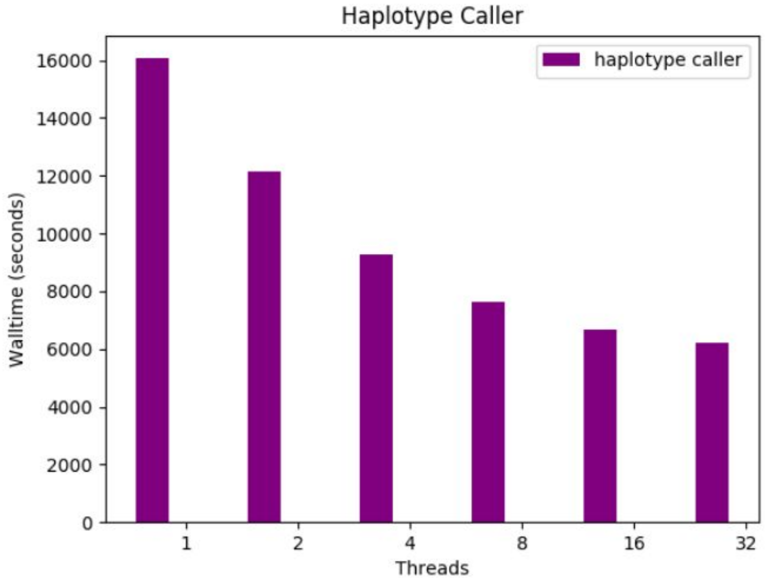
1. Differences across cancers
2. Cancer epidemiology and health disparities
3. Therapeutics opportunities in precision medicine

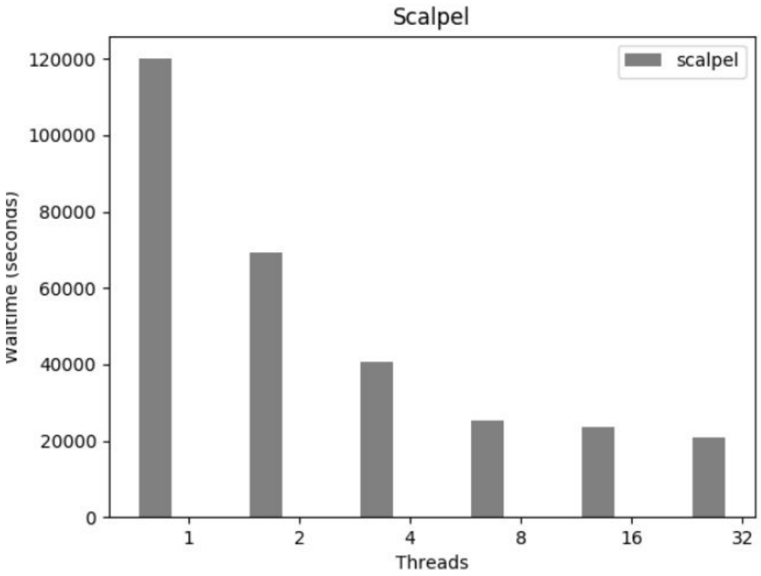
SwiftSeq & Swag (Software)
Baughman, M. IEEE UCC 2018
Wu, C. CloudCom 2019

Wu, A. *Brief Bioinform* 2023
Perera, A. *Commun Biol* 2024

Pitt, JJ. *Nat Commun* 2018
Wang, C., Pitt, JJ. *Int J Cancer* 2019
Nasari-Pour, N. *Nat Commun* 2021
Wai, C. *Haematologica* 2022
Kong, L. *Cell* 2024

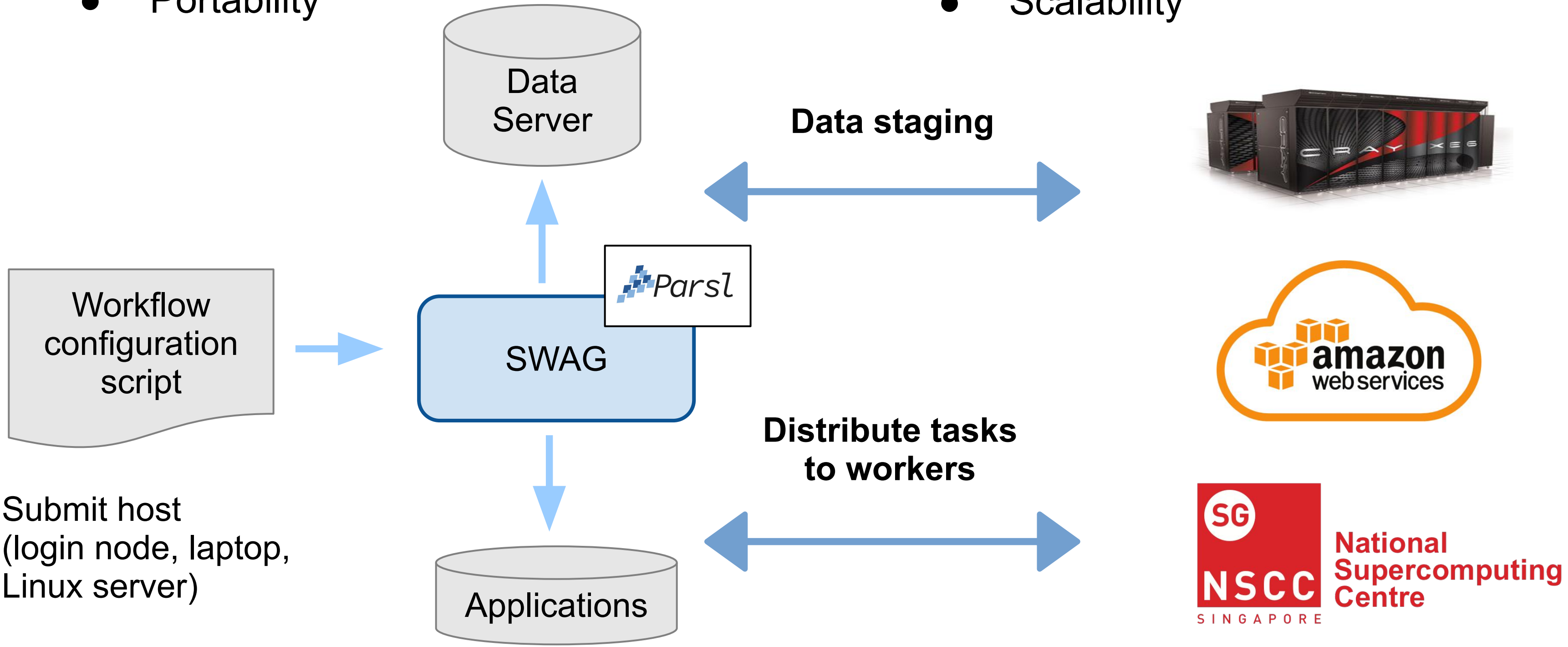
Non-linear Multithreading of Bioinformatics Tools





SWAG: Scalable Workflows for Analyzing Genomes

- Transparent parallelism
- Portability
- Robust to failures
- Scalability



Data Server

Data staging

Parsl

Workflow configuration script

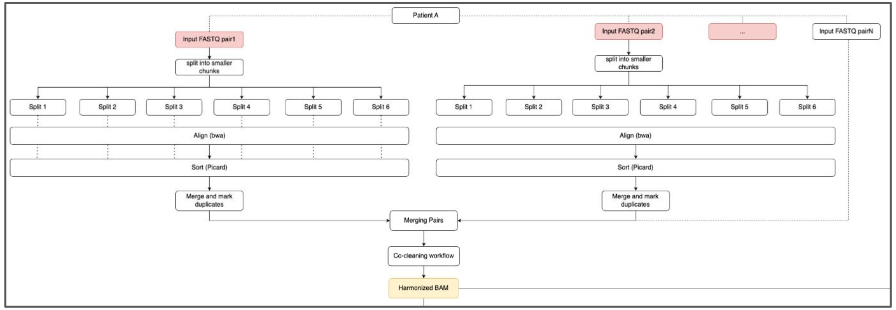
SWAG

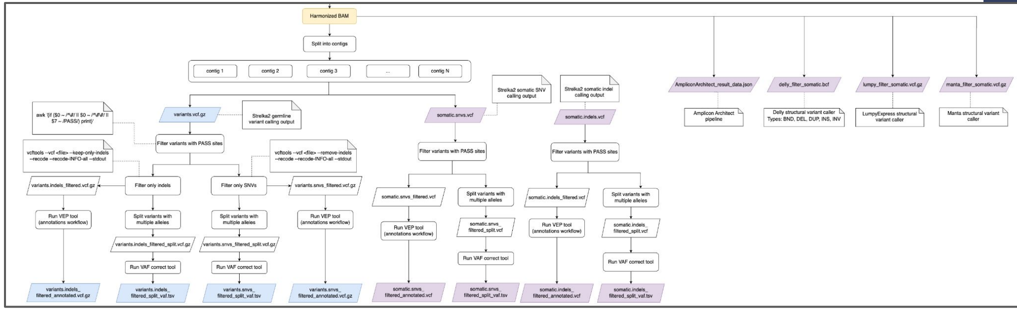
Distribute tasks to workers

Submit host (login node, laptop, Linux server)

Applications

SWAG - GDC Workflow

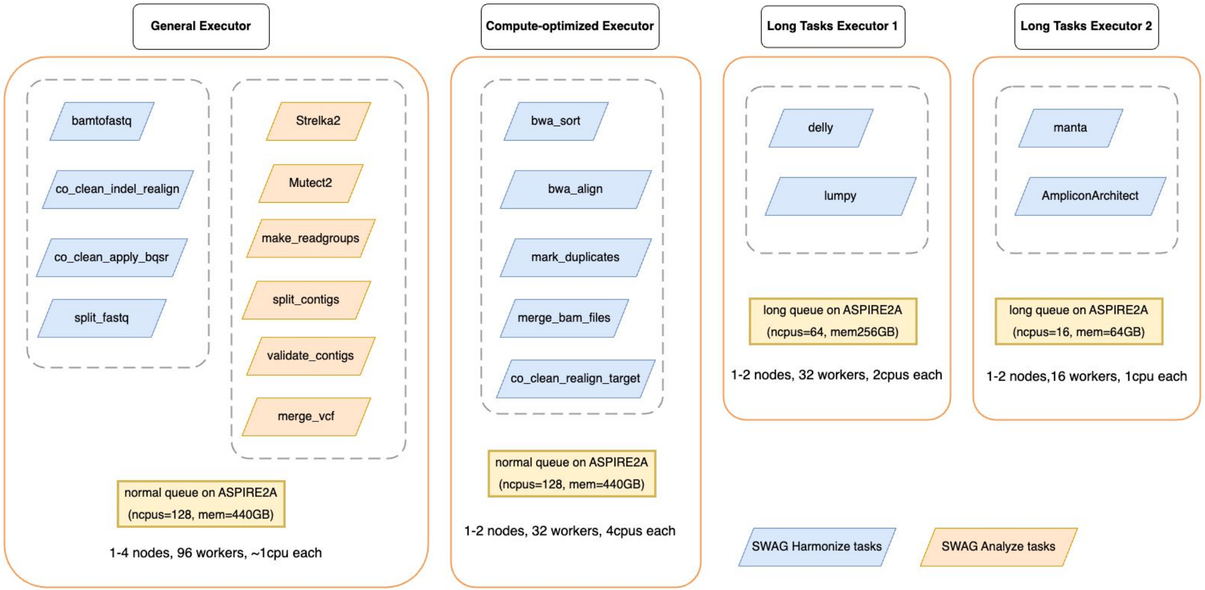




Harmonized BAM
Split into contigs
contig 1
contig 2
contig 3
...
contig N
awk '{if ($0 ~ /^#/ || $0 ~ /^#/ || $7 ~ /PASS/) print}'
variants.vcf.gz
Strelka2 germline variant calling output
Filter variants with PASS sites
vcftools --vcf <file> --keep-only-indels --recode --recode-INFO-all --stdout
vcftools --vcf <file> --remove-indels --recode --recode-INFO-all --stdout
variants.indels_filtered.vcf.gz
Filter only indels
Filter only SNVs
variants.snvs_filtered.vcf.gz
Run VEP tool (annotations workflow)
Split variants with multiple alleles
Split variants with multiple alleles
Run VEP tool (annotations workflow)
variants.indels_filtered_split.vcf.gz
variants.snvs_filtered_split.vcf.gz
Run VAF correct tool
Run VAF correct tool
variants.indels_filtered_annotated.vcf.gz
variants.indels_filtered_split_vaf.tsv
variants.snvs_filtered_split_vaf.tsv
variants.snvs_filtered_annotated.vcf.gz
Strelka2 somatic SNV calling output
Strelka2 somatic indel calling output
somatic.snvs.vcf
somatic.indels.vcf
Filter variants with PASS sites
Filter variants with PASS sites
somatic.snvs_filtered.vcf
Split variants with multiple alleles
somatic.indels_filtered.vcf
Split variants with multiple alleles
Run VEP tool (annotations workflow)
somatic.snvs_filtered_split.vcf
Run VEP tool (annotations workflow)
somatic.indels_filtered_split.vcf
Run VAF correct tool
Run VAF correct tool
somatic.snvs_filtered_annotated.vcf
somatic.snvs_filtered_split_vaf.tsv
somatic.indels_filtered_annotated.vcf
somatic.indels_filtered_split_vaf.tsv
AmpliconArchitect_result_data.json
delly_filter_somatic.bcf
lumpy_filter_somatic.vcf.gz
manta_filter_somatic.vcf.gz
Amplicon Architect pipeline
Delly structural variant caller Types: BND, DEL, DUP, INS, INV
LumpyExpress structural variant caller
Manta structural variant caller

- Genomic raw fastq data split into multiple chunks for parallelization
- BAM files split by region for parallel variant calling
- Compute considerations due to chromosome size variability
  - Dynamic resource allocation
  - Multiple executors with different walltimes supported by underlying queues

SWAG - Optimizing Workflow Performance



General Executor

bamtofastq

co_clean_indel_realign

co_clean_apply_bqsr

split_fastq

Strelka2

Mutect2

make_readgroups

split_contigs

validate_contigs

merge_vcf

normal queue on ASPIRE2A (ncpus=128, mem=440GB)

1-4 nodes, 96 workers, ~1cpu each

Compute-optimized Executor

bwa_sort

bwa_align

mark_duplicates

merge_bam_files

co_clean_realign_target

normal queue on ASPIRE2A (ncpus=128, mem=440GB)

1-2 nodes, 32 workers, 4cpus each

Long Tasks Executor 1

delly

lumpy

long queue on ASPIRE2A (ncpus=64, mem256GB)

1-2 nodes, 32 workers, 2cpus each

Long Tasks Executor 2

manta

AmpliconArchitect

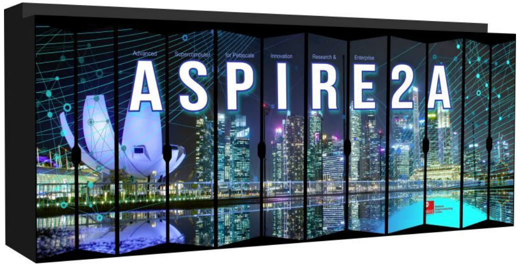
long queue on ASPIRE2A (ncpus=16, mem=64GB)

1-2 nodes,16 workers, 1cpu each

SWAG Harmonize tasks

SWAG Analyze tasks

ASIPRE2A – Singapore's National
Petascale Supercomputer
(~800 nodes & ~100,000 cores)



- Canned configurations for Parsl executors for each workflow
- Easily to switch between various HPC/compute infrastructure
- Task profiling to understand resource utilization patterns
- Dynamic assignment of tasks to executors based on scalability profile

Key Results to Date

- Processed 250+ whole genome sequencing (WGS) of isogenic cell lines and mouse tissue samples
  - ~164,000 cpu core hours on ASPIRE1 (NSCC)
  - ~50 tb of raw data
- Processed 12,000+ whole exome sequencing (WXS) samples from TCGA
  - ~2M cpu core hours on ASPIRE1 (NSCC)
  - ~800 tb of raw data
- Processed 250+ deeply sequenced WGS samples from TCGA
  - ~200,000 cpu core hours on ASPIRE2A (NSCC)
  - ~250 tb of raw data

Related manuscripts

Kong LR… Pitt JJ, Venkitaraman AR. *Cell* 2024

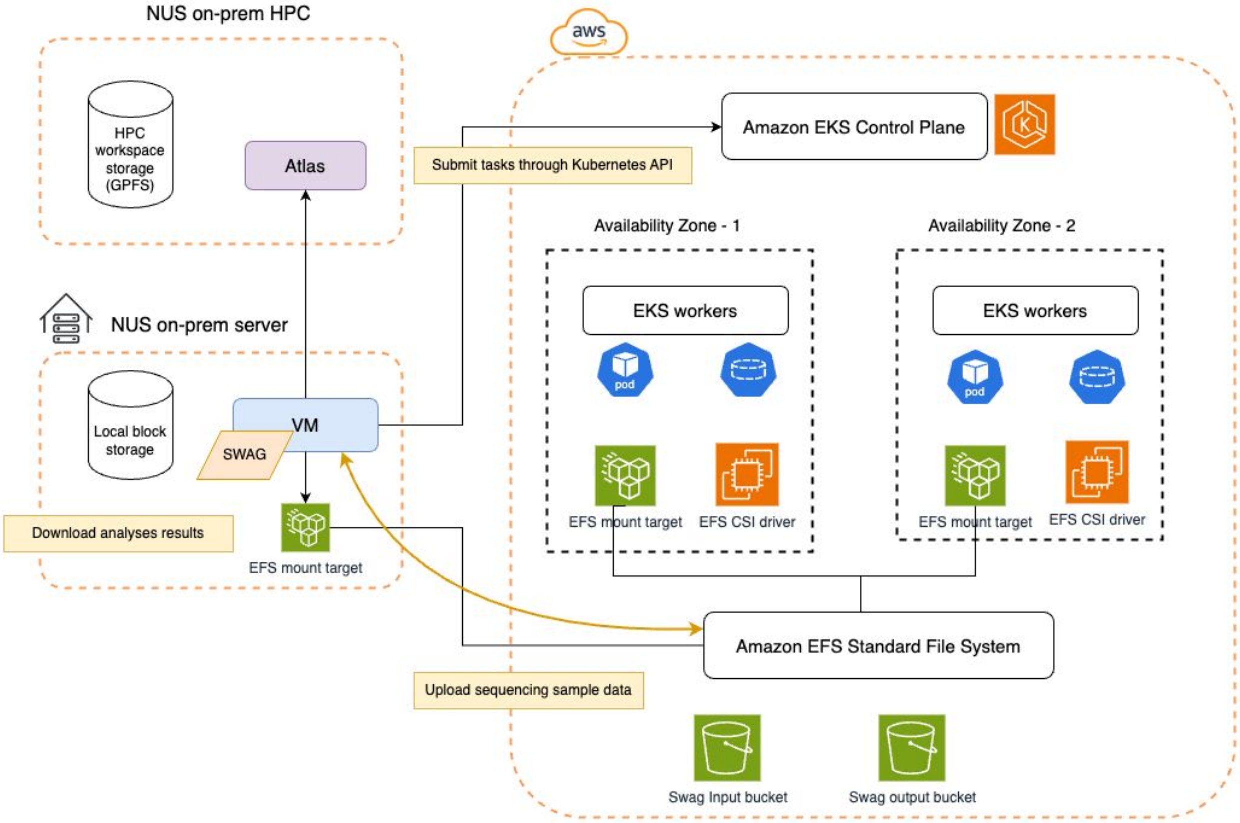
Perera A... Pitt JJ. *Communications Biology* 2024

Wu A… Pitt JJ. *Briefings in Bioinformatics* 2023

Wong H… Pitt JJ. *In preparation*

Wu A… Pitt JJ. *In preparation*

SWAG Roadmap - Multi-site Genomic Workflows



Improving Pipeline Efficiency by Advancing Parsl Capabilities

- Task failures due to biological variability
  - Chromosomes vary widely in length
  - WGS sample data size varies from 100Gb - 300Gb
  - Variability in runtime causes task timeouts
- Inefficient distribution of tasks
  - Whole node allocation requirement for consistent performance in HPC
  - Resource waste due to idling cpu cores
  - Guaranteed failure when allocated to an old node

- Dynamic assignment of tasks
  - Duration-sensitive task allocation
  - Feedback loop through a user provided function on retries
- Flexible task allocation to reduce costs
  - Cost-effective task packing strategy
  - Node-aware scheduling