Performance Optimization For WGS/WES

Tao Yang

mike.yang@foxmail.com

Oct, 2018

The definition of Performance

- This document focus on reduce the execution time for one WGS/WES analysis pipeline. The performance means pipeline **process latency**.
- Running multiple WGS/WES analysis pipelines at the same time on different CPU core, also can increase the process throughput on single server system. Pipeline process throughput is **NOT** the target for this document.

Key methods for optimization

- Parallel workload to make full utilization of multi-core CPU resource.
    - Modify the bowtie2 and samtool parameter to match the CPU core count in the system.
    - Use the GATK SPARK mode to run GATK on multi core.
- Reduce the IO waiting time and IO operation
    - Use the SSD to replace the HDD
    - Merge multi stage pipeline into single stage pipeline, remove the overhead of writing data to disk and loading the data from disk.
- Use CPU vector instruction (AVX 512)
    - GATK 4.0 docker image doesn’t include Intel AVX 512 library. Use GATK native mode can use the Intel AVX library.
- Reduce the system overhead
    - Cent OS 7.4/7.5 include a patch for security protection, which have performance impact on IO workload. Disable it with Linux grub parameter “spectre_v2=off nopti”
- Use the right hardware
    - Use Intel Xeon CPU instead of Intel Xeon Phi CPU.
    - Use FPGA to accelerate some algorithm, such as PairHMM.

Performance data

	Xeon Phi 7250 HDD	Xeon 8170 HDD	Xeon 8170 HDD spectre_v2=off	Xeon 6138 HDD spectre_v2=off	Xeon 8170 SSD	Xeon 8170 SSD spectre_v2=off	Xeon 6138 SSD spectre_v2=off	Xeon 8170 HDD SPARK	Xeon 8170 SSD SPARK	Xeon 8170 SSD SPARK spectre_v2=off	Xeon 6138 SSD SPARK spectre_v2=off
bowtie2+samtools sort	1:43:15	0:19:29	0:15:11	0:21:48	0:19:03	0:13:11	0:20:14		0:19:16	0:14:45	0:19:37
MarkDuplicates	1:28:02	0:26:52	0:25:38	0:24:16	0:29:17	0:24:33	0:22:52		0:10:14	0:10:12	0:10:09
AddOrReplaceReadGroups	0:27:29	0:05:34	0:06:35	0:05:59	0:05:42	0:05:48	0:05:39		0:00:00	0:00:00	0:00:00
BaseRecalibrator	3:10:52	0:25:10	0:22:36	0:23:02	0:24:09	0:22:31	0:23:00		0:00:00	0:00:00	0:00:00
ApplyBQSR	2:51:47	0:21:07	0:19:45	0:20:21	0:21:06	0:18:55	0:19:35		0:11:32	0:11:15	0:11:48
HaplotypeCaller	8:39:11	1:39:01	1:18:58	1:26:26	1:38:43	1:19:43	1:28:31		0:12:53	0:11:30	0:13:28
Total time	**18:20:36**	**3:17:13**	**2:48:43**	**3:01:52**	**3:18:00**	**2:44:41**	**2:59:51**	**2:00:00**	**0:53:55**	**0:47:42**	**0:55:02**

- Need to analysis why Xeon + SDD only have very little performance improve compare with HDD
- GATK 3.0 use multi thread mode for multi core CPU. GATK 4.0 use SPARK mode.
- GATK SPARK result have 3% variant less than non SPARK version. GATK community is working on it

Platform Information

	Xeon(R) 8170 @ 2.10GHz	Xeon Phi 7250 @1.4Ghz
CPU	2* 26 Core / 2*52 Threads	68 Core / 272 Threads
Memory	128G DDR4 2666	256G DDR4 2400
SSD	Intel NVMe P4500 1TB	
HDD	WD 1T HDD	WD 8T HDD
OS	CentOS 7.5	CentOS 7.4
CPU Frequency	2.1Ghz	1.4Ghz
CPU Turbo	3.7Ghz	1.6Ghz
CPU Cache	35.75 MB L3 + 26 MB L2	34 MB L2
	A few big cores, good for the application need single core performance.	Many small cores, good for the full paralleled application.

* All the test is running on Xeon 8170
Xeon 8160 @2.1G with 24 cores may have better perf/$ than Xeon 8170

What is next?

- Following the GATK community on the SPARK version bug fix.
- Use FPGA to accelerator HaplotypeCaller
- Do a cost analysis of the FPGA card.