# CPU_GPU_ASSIGNMENT_1.1

*by* ASHISH VIKRAM SINGH

**Submission date:** 14-Sep-2019 02:48PM (UTC+0100)
**Submission ID:** 1172062913
**File name:** CS7NS1-A-SEM101-201920_83395911_1091770171_CPU_GPU_ASSIGNMENT_11.pdf (481.33K)
**Word count:** 408
**Character count:** 2273

**"Speedup and Power Scaling Models for Heterogeneous Many-Core Systems"**

**a)**

- This paper gives overview about heterogeneous computing which are multicore in nature, a mechanism which combines different processors of different kinds.
- Keeping intact older methods for efficiency and performance of the homogenous architectures and improving it also facing its challenges.
- Power management and load balancing improvements using these cores.
- Some combination of heterogenous models have shown bad performance results as well.

**b)**

- By using the "load balancing models" results are of much improved standards.
- The performance of the cores is enhanced, and utilization of the core is also maximised.
- It gives a Comparison and brief insight of the heterogenous and homogenous models
- the tests are executed with real life scales and their limitations.

**c)**

- use of accelerators such as "Open Cl" with the devices having different instruction set.
- The core scaling is perfectly linear using CPU's and GPU's
- With proper scaling and load balancing the performance is increased up to 400% in some models.
- The blending of CPU's and GPU's have shown significant good results in terms of power and energy dissipation.

**"Scalable GPU Virtualization with Dynamic Sharing of Graphics Memory Space."**

**a)**

- The hardware is restricted when it comes to graphics in cloud environment and its memory so in order to enhance a solution called "gScale" is introduced.
- Various slots and scheduling mechanism division of HD graphics memory.
- "The private shadow GTT" it authorizes the memory to be more distributable between the vGPU's
- a technique called "ladder mapping and fence memory space pool" that permits" CPU's graphic memory" to access an alternative route for the graphics memory space that are global.

**b)**

- The slot splitting and sharing method is used to tweak the efficiency of vGPU's.
- In cloud environment it improves the predictive GTT copy mechanism by accelerating it. This is called predictive copy aware scheduling
- "Fine grained slotting" helps in dividing the memory particularly related to graphics into sub slots hence giving the cloud vendors a more adaptable arrangement interface.
- "predictive GTT copy mechanism"- it increases the efficiency by swapping "private shadow GTT" ahead of context switching.

**c)**

- gVirt has a scalability issue wherein it restricts the guest's vGPUs occasions. Hence "gScale" on top of gVirt therefore making it more scalable.
- Increase in the efficiency and performance of vGPU, 5 times in linux and 4 times in Windows
- efficient load balancing among the cloud space
- "gScale" has a slight runtime advantage when accommodating multiple vGPU's instances.

# CPU_GPU_ASSIGNMENT_1.1

ORIGINALITY REPORT

| 0% | 0% | 0% | 0% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

1%

★ www.computer.org

Internet Source

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 2 words |
| Exclude bibliography | On | | |