

# Development of Hyperset Paging mechanism

Akhter Hussain (✉ [pral.hohai@gmail.com](mailto:pral.hohai@gmail.com))

Comsats Institute of IT <https://orcid.org/0000-0002-6785-9480>

---

## Research Article

**Keywords:** paging, memory management, indexing, segmentation

**Posted Date:** April 22nd, 2020

**DOI:** <https://doi.org/10.21203/rs.3.rs-23970/v1>

**License:**   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# Abstract

Algorithms for page replacement play important roles in virtual memory management, especially in paging operating systems. Page substitution happens when the required page is not retained in the memory (page fault) or the free accessible memory is not adequate to fulfill the requirement. It is either there are none or the amount of free sites is fewer than the required total. Two regular page replacements were hybridized as Hybrid Page replacements (HRA) in this analysis (LRU and Optimal/OPT algorithms recently used). For its service, HRA is based on the principle of OPT and LRU. The HRA was determined by comparing the page failures caused with the default algorithms First In First Out (FIFO), LRU and Optimal. Tests showed the amount of frames through to 4 and beyond the HRA outperformed FIFO, OPT and LRU.

## I. Introduction

In the operating system, the paging replacement mechanism is crucial. Page replacement algorithms allow the decision from which pages the memory will be replaced if the required frame is not accessible. But to solve this problem, some algorithm was created. The algorithms differ from each other in the way in which the page removal process is been handled (Genta, 2015). All page replacement algorithms are internally similar to the following: insertion, detection and searching for a page. However, most of these algorithms depends on specialized data structure for effective performance (Kavaraid, 2013). Most of the algorithms handle huge amount of data, the locality of reference is considered as a shared attribute between programs. Hence, the majority of applications does not access all data at the same time. Instead of this, they reference only small part of data at different point in time (Shen et al, 2004). The locality of reference can be adopted in two different approaches: spatial locality of reference and temporal locality of references. The spatial locality is of the view that nearby memory location will be referenced in the near future while temporal locality depends on the time the page spends in the frame (O'neil *et al.*, 1993).

The localities of reference can be increased by careful selection of data structure used in the algorithm. The data structure will reduce the page fault rate as well as the number of pages in working set. For example, a stack has high locality because the replacement algorithm of the stack always accesses the top. Some measures: memory reference, search speed and a total number of page touched are used to measure the performance of page replacement algorithms (Khulbe et al, 2014). The efficiency of page replacement algorithm is an open question for the study since there are several factors that affects the performance of an algorithm (Brinkhoff, 2002). The memory access has a clear impact on page replacement algorithm performance (Anthony, 2015).

To analyze the performance of page memory system, some page frame number (PFNs) has to be generated during the execution of a given program (Hwang and Jotwani, 2011). Several properties should cover the effective replacement algorithm, it should be able to distinguish the hot and cold resident in cache to reduce the number of page faults. Furthermore, efficient algorithm implementation can be

achieved if it uses a constant and small portion of the memory that stores page history in the cache, it should not consume large memory space. Generating minimum number of page fault implies better performance and this study proposed the hybrid page replacement algorithm which is a combination of the OPT and LRU page replacement algorithms. The performance of the hybrid system will be compared with the existing page (FIFO, LRU and OPT) replacement algorithms.

## ii. Related Works

Park, *et al.*, (1986), worked on Clean First Least Recently Used (CFLRU). It was the first page replacement algorithm ever proposed for the NAND flash memory. This was initially designed in consequence of the unique hardware constraints posed by the NAND flash storage memory. The algorithm was based on the original LRU algorithm and was enhanced to fit the needs of the operating system. The concept of cold and hot pages was introduced on the existing Clean and Hot pages. The CFLRU considered the entrance recurrence of every page inside of the cold-first regions of the clean page rundown and dirty page list. The performance of CFLRU algorithm was compared with those of LRU and the CFLRU algorithm demonstrated the most noteworthy page hit ratio. Jung, *et al.*, (2000), proposed Least Recently Used-Write-Sequence-Reordering (LRU-WSR). The proposed LRU-WSR was a modified version of LRU introduced with Write Sequence Reordering (WSR) strategy. The LRU-WSR algorithm maintains a single page list in the LRU order similar to the original LRU algorithm. However, any dirty pages in the list will have a cold flag which may or may not be set as per the situational operation. The identification of Cold or Hot of a dirty page is taken care of by a cold detection algorithm. A dirty page may be cold, if that is the least referenced page in the list and its cold flag is set. A dirty page may be hot if that page is referenced again and the cold flag on it is cleared. Results from this work was compared with FIFO and found 18% better off than FIFO. Mark Page Discard Randomly Caching Algorithm (MPDRC) was proposed by Sumit *et al.*, (2015), the work was developed in windows XP using C# .Net and the unique identification number was assigned to unique URLs to log of proxy server. These numbers were taken as a reference string that suits input to the innovative algorithms. The focus of the research was exploration of web proxy caching algorithm best for proxy server. With the help of log in details of proxy server, a real trace web reference was obtained. For the simulation, each URL was given a numeric identity which enables the numeric reference string to be acquired. During simulation the author compared MPDRC with FIFO, LRU and LFU replacement algorithms with the proposed innovative page replacement algorithm (MPDRC). It was discovered that MPDRC algorithm has 8.34% better than FIFO, 16.63% better than LFU and 10.6% better than LRU algorithm. Overall average this work innovative page replacement examined 11.85% better than existing algorithm. After comprehensive simulation experiments is summarized that for proxy caching the MPDRC hit ratio performance better than others existing algorithms. A study by Paajanen (2000), worked on Least Recently Used (LRU). The algorithm takes into considered the imbalance of the cost of read and writes memory when replacing pages. The algorithm defines that, the string been used in the list was replaced in the needed frame. The LRU algorithm has better performance when compared with FIFO.

## iii. Method

In this study, three standard algorithms were considered; FIFO, LRU and OPT

- i. **First in First out (FIFO):** It replace the page at the head of the queue when a page is brought into memory. FIFO associates with each page, the time the page was brought to the tail of the queue (Abraham, et al., 2012).
- ii. **Optimal:** In operating system the algorithm swaps out the page whose next reference will take longer time for future reference. For example, a page that is not going to be used for the next 6 seconds will be swapped out over a page that is going to be used within the next 0.4 seconds(Khulbe, et al., 2014).
- iii. **Least Recently Use (LRU):** This based on the probability theories in a logic idea. This idea consists of pages which have been heavily used in the last few instructions, which will probably be used again in the next few pages that are not use for a long time may say in this state again.

The following are some of the lapses observed with the existing algorithms

- i. The optimal algorithm fetches the string that will be used in the future, whereas the operating system cannot predict the future occurrence. Therefore, the optimal is not a good choice
- ii. FIFO algorithms fetch and replace the string that came into the frame first. This operation is based on first in first out (queue). As a result of this operation, more page fault is generated. Hence, FIFO is not a good choice.
- iii. LRU, this algorithm is similar to FIFO just that it fetches in the page that has been used the least, this is better than FIFO but still have high page fault than OPTIMAL.

### Hybrid Page Replacement Modeling

The HRA comprises of two standard page replacement algorithms namely, LRU and Optimal. These where hybridized to give a better performance in terms of number of page faults generated when compared to the individual paging algorithm. As shown in fig.1, the proposed HRA algorithm is designed for the purpose of less page fault in the page replacement algorithm. Knowledge of LRU and OPT algorithm is crucial in the usage of HRA algorithm. The victim page is been swapped in/out of the physical to the backing store (virtual memory) using HRA technique. The frame is shared between OPT and LRU i.e. the first frame is given to OPT, and it takes the first reference string, processed it with the frame and swap out the particular frame that will not be used for the longest period of time and LRU will pick the second reference string, process it with the frames and swap out the particular frame that has been used least recently among all the frames on the pages. Java SE was used to implement the algorithm and the units that were tested includes: the first dialog box for the input frame, the second for the length of the strings and the third for the reference strings itself.

## IV. Results And Discussion

In this study, the results achieved are the outcome of simulation of the designed system. This result is the evaluation performance metric of the proposed HRA system in terms of NUMBER OF PAGE FAULTS generated. Fig. 2 depicts the simulation interface where the experiments were conducted. Three (3) cases with reference strings (70120304230321201701, 470710121271 and 12341251234) were considered with 3,4,5,6 and 7 frames on each.

**Case 1:** In this case, reference string 70120304230321201701 was simulated on 3, 4, 5, 6 and 7 frames. The result in fig. 3 depicts the performance of HRA when compared with FIFO, LRU and OPT. It shows that HRA outperformed among others in terms of lower number of page fault generated as the frame number increases from 4 and above.

**Case 2:** In this case, reference string 470710121271 was simulated on 3, 4, 5, 6 and 7 frames. The results in fig. 4 depicts the performance of HRA when compared with FIFO, LRU and Optimal. It was observed that, with few reference strings, HRA have similar performance with OPT but still outperformed OPT from 4 frames and above.

**Case 3:** In this case, reference string 123412512345 was simulated on 3, 4, 5, 6 and 7 frames. The results in fig. 8 shows that HRA have poor performance at the lower number of frame, but from 4 frames and above, it outperformed others in terms of lower number of page fault generated.

## V. Conclusion And Future Works

Developed and tested an efficient page substitution algorithm called HRA, which used LRU and OPTIMAL algorithm concepts. Via multiple experiments, HRA models have been found to be equivalent to the current Operating System page substitution by default link. Extensive work will be performed to develop HRA on smaller frames.

## References

1. Ahmed, S.H.; Lee, S.; Park, J.; Kim, D.; Rawat, D.B. iDFR: Intelligent directional flooding-based routing protocols for underwater sensor networks. (CCNC), Las Vegas, NV, USA (2017); doi.org/10.1109/ccnc.2017.7983168.
2. Yang, H., Zhou, Y., Hu, Y. H., Wang, B., & Kung, S. Y. (2018, May). Cross-layer design for network lifetime maximization in underwater wireless sensor networks. In *2018 IEEE International Conference on Communications (ICC)* Doi: 10.1109/ICC.2018.8422176
3. Coutinho, R. W. L., Boukerche, A., Vieira, L. F. M., & Loureiro, A. A. F. (2018). Underwater Wireless Sensor Networks. *ACM Computing Surveys*, 51(1), 1–36. <https://doi.org/10.1145/3154834>
4. Somani, Arun K., et al. *Smart Systems and IoT: Innovations in Computing: Proceeding of SSIC 2019*. Springer, 2020.
5. Qureshi, U., Shaikh, F., Aziz, Z., Shah, S., Sheikh, A., Felemban, E., & Qaisar, S. (2016). RF Path and Absorption Loss Estimation for Underwater Wireless Sensor Networks in Different Water

- Environments. *Sensors*, 16(6), 890. <https://doi.org/10.3390/s16060890>
6. Tilwari, V., Maheswar, R., Jayarajan, P., Sundararajan, T. V. P., Hindia, M. N., Dimyati, K., Amiri, I. S. (2020). MCLMR: A Multicriteria Based Multipath Routing in the Mobile Ad Hoc Networks. *Wireless Personal Communications*. <https://doi.org/10.1007/s11277-020-07159-8>
  7. Ashraf, S., Ahmed, T., Raza, A., & Naeem, H. (2020). Design of Shrewd Underwater Routing Synergy Using Porous Energy Shells. *Smart Cities*, 3(1), 74–92. <https://doi.org/10.3390/smartcities3010005>
  8. Qu, J., Zhang, Z., Cui, Y., Wang, J., & Mastorakis, G. (2019). Research and Application of Multi-Node Communication and Energy Consumption Prediction Control in Underwater Acoustic Network. *IEEE Access*, 7, 41220–41229. doi:10.1109/access.2019.2907376
  9. Ashraf, S., Gao, M., Chen, Z., Kamran, S., & Raza, Z. (2017). Efficient Node Monitoring Mechanism in WSN using Contikimac Protocol. *International Journal of Advanced Computer Science and Applications*, 8(11). doi:10.14569/ijacsa.2017.081152
  10. Su, Y., Fan, R., Fu, X., & Jin, Z. (2019). DQELR: An Adaptive Deep Q-Network-Based Energy- and Latency-Aware Routing Protocol Design for Underwater Acoustic Sensor Networks. *IEEE Access*, 7, 9091–9104. doi:10.1109/access.2019.2891590
  11. Guan, Q., Ji, F., Liu, Y., Yu, H., & Chen, W. (2019). Distance-Vector-Based Opportunistic Routing for Underwater Acoustic Sensor Networks. *IEEE Internet of Things Journal*, 6(2), 3831–3839. doi:10.1109/jiot.2019.2891910
  12. Vijayalakshmi, P.; Noorunnisa, N.; Rajendran, V. Performance analysis of VBF protocol for position tracking of moving nodes in underwater communications. In *Proceedings of the 2017 International Conference (ICCSP)*, Wuhan, China, 17-19 March 2017; doi:10.1109/iccsp.2017.8286708.
  13. Khasawneh, A.; Latiff MS, B.A.; Kaiwartya, O.; Chizari, H. A reliable energy-efficient pressure-based routing protocol for underwater wireless sensor network. *Wirel. Netw.* 2017, 24, 2061–2075, doi:10.1007/s11276-017-1461-x.
  14. Tran-Dang, H.; Kim, D.S. Channel aware cooperative routing in underwater acoustic sensor networks. *J. Commun. Netw.* 2019, 21, 33–44, doi:10.1109/JCN.2019.0000004.
  15. Daudpota, S. (2019). A Comprehensive Survey on the Performance Analysis of Underwater Wireless Sensor Networks (UWSN) Routing Protocols. *International Journal of Advanced Computer Science and Applications*, 10(5). <https://doi.org/10.14569/ijacsa.2019.0100576>.  
<https://doi.org/10.13140/RG.2.2.13755.67363>
  16. Ashraf, S., Gao, M., Mingchen, Z., Ahmed, T., Raza, A., & Naeem, H. (2020). USPF: Underwater Shrewd Packet Flooding Mechanism through Surrogate Holding Time. *Wireless Communications and Mobile Computing*, 2020, 1–12. doi:10.1155/2020/9625974
  17. Wang, Z.; Han, G.; Qin, H.; Zhang, S.; Sui, Y. An Energy-Aware and Void-Avoidable Routing Protocol for Underwater Sensor Networks. *IEEE Access* 2018, 6, 7792–7801, doi:10.1109/access.2018.2805804.
  18. Hindu, S.K.; Hyder, W.; Luque-Nieto, M.-A.; Poncela, J.; Otero, P. Self-Organizing and Scalable Routing Protocol (SOSRP) for Underwater Acoustic Sensor Networks. *Sensors* 2019, 19, 3130,

doi:10.3390/s19143130.

19. Barbeau, M.; Blouin, S.; Cervera, G.; Garcia-Alfaro, J.; Kranakis, E. (2015). Location-free link state routing for underwater acoustic sensor networks. In Proceedings of the 2015 IEEE 28th Canadian Conference(CCECE), Halifax, NS, doi:10.1109/ccece.2015.7129510.
20. Balsamo, S., Marin, A., & Vicario, E. (Eds.). (2018). New Frontiers in Quantitative Methods in Informatics. Communications in Computer and Information Science. Springer International Publishing. doi:10.1007/978-3-319-91632-3
21. Balas, Valentina Emilia., et al. Data Management, Analytics and Innovation: Proceedings of ICDMAI 2018, Volume 1. Springer Singapore, 2019.

## Figures

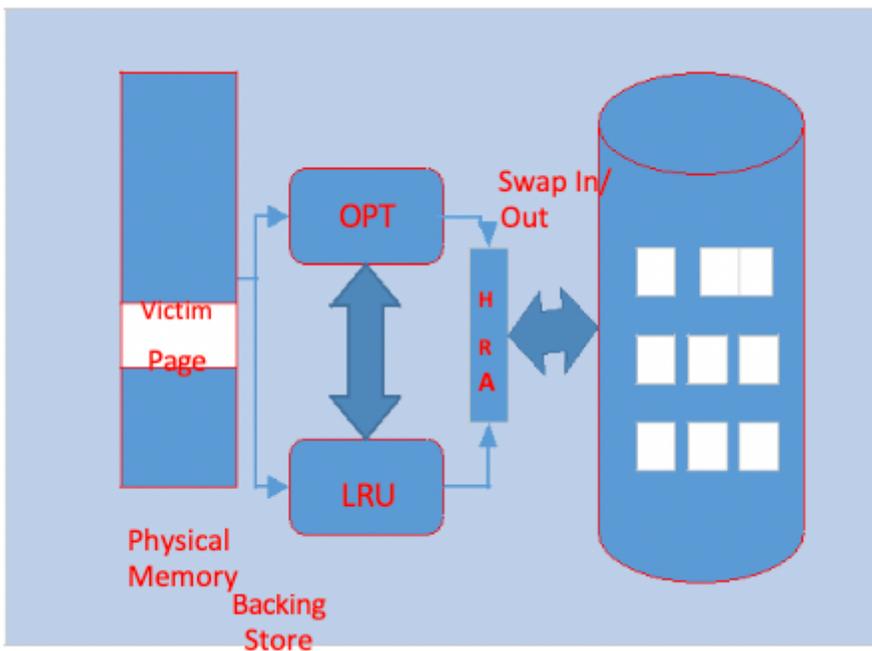


Figure 1

Proposed HRA Model

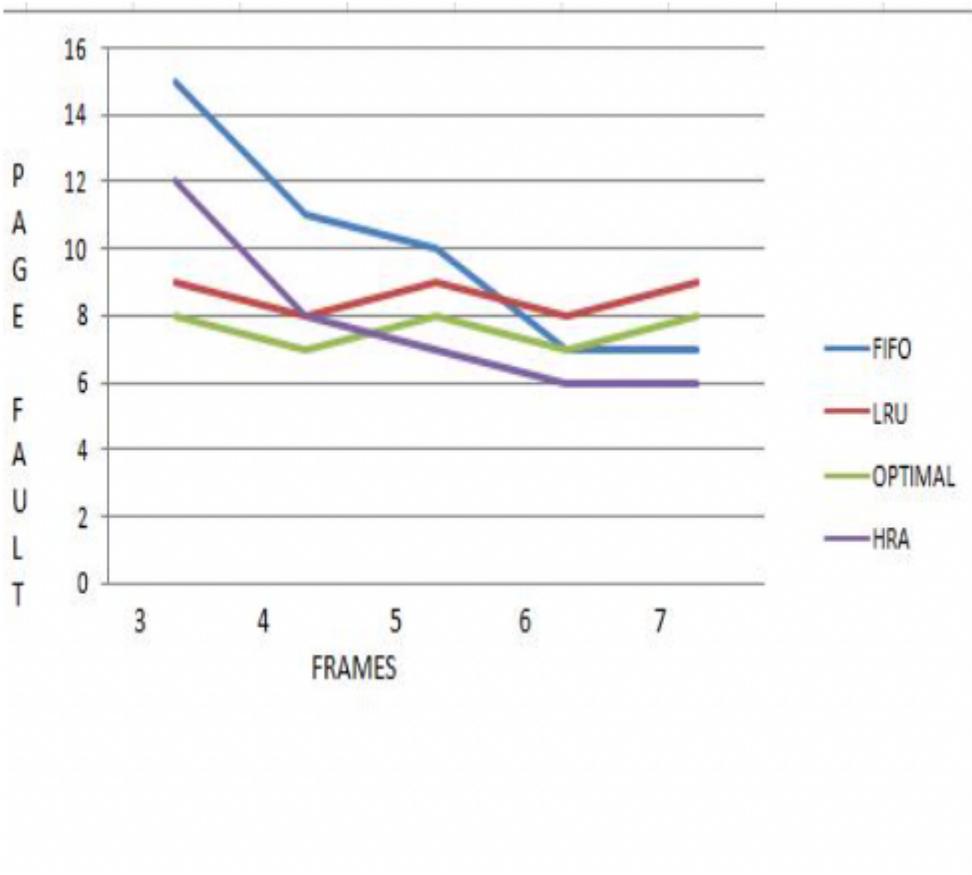
```
Total number of page fault : 6
Do you want to continue? if Yes, press 1

Menu
1.FIFO
2.LRU
3.Optimal
4.hybrid
5.Exit
Enter your choice : 4

String is : 1
String is : 2
String is : 3
String is : 4
String is : 1
String is : 2
String is : 5
String is : 1
String is : 2
String is : 3
String is : 4
String is : 5
Total number of page fault : 5
Do you want to continue? if Yes, press 1
```

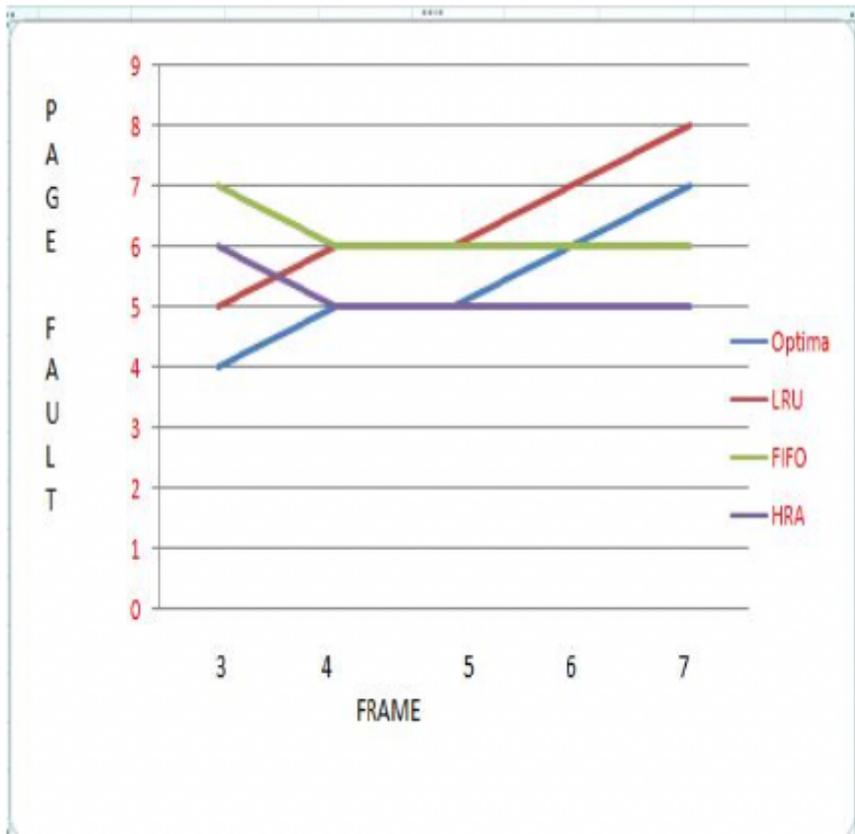
Figure 2

simulation console



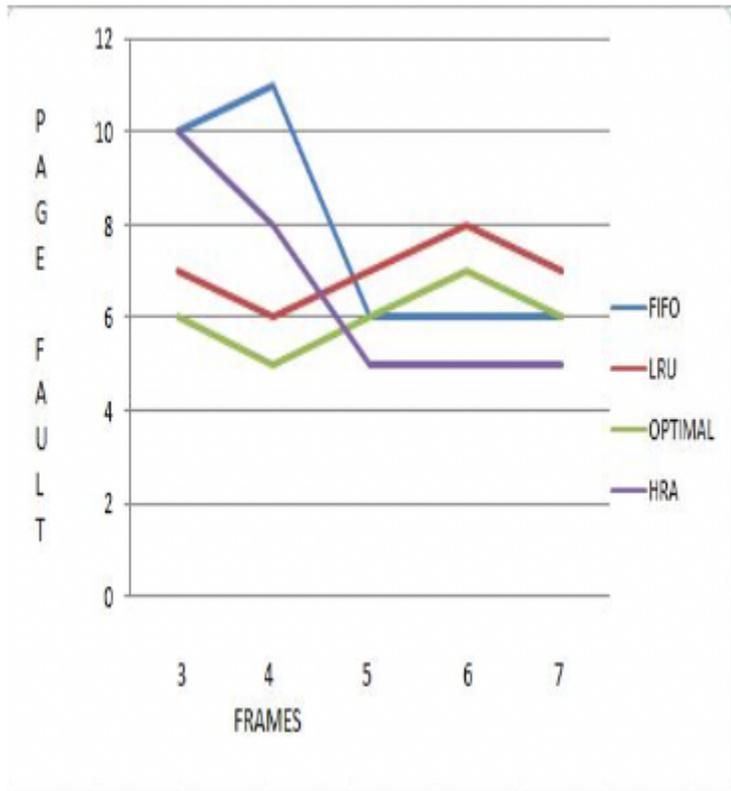
**Figure 3**

Graph of HRA with 20 reference strings.



**Figure 4**

HRA with 12 reference strings.



**Figure 5**

HRA with 12 reference strings.

**Figure 6 was omitted by the authors in this version of the paper.**

**Figure 6**

**Figure 7 was omitted by the authors in this version of the paper.**

**Figure 7**

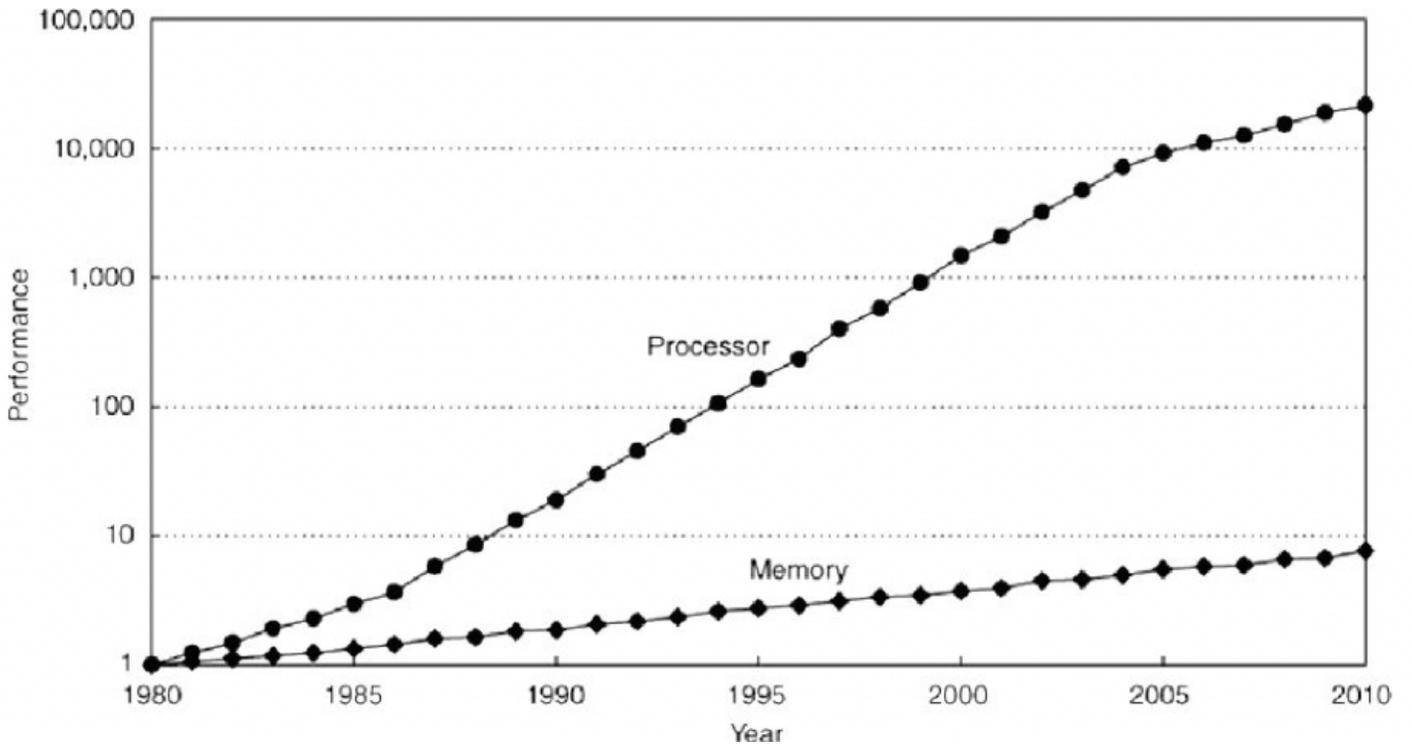


Figure 8

A legend was omitted by the authors in this version of the paper.