Bernie Piece Acceptance Speech for CMG’s A.A. Michelson Award

The following text is the acceptnce speech of Bernie Pierce after being recognized in 1998 for his many contributions to the measurement and performance of computer systems. For more information about the A.A.Michelson award, please visit the Computer Measurement Group website at www.cmg.org.

This text is published with the permission of June Pierce and the Pierce family.

CMG President Thomas Dennison awarding Bernie Pierce the 1998 A.A. Michelson award

I would like to express my appreciation to all those who participated in selecting me as a recipient of this award. I am truly humbled when I look around at those who have been honored before me. I have worked in the area of Computer Performance for over two decades at IBM and Candle. The majority of this time was spent working with the mainframe operating system, MVS and its predecessors MVT and SVS. The decision to become a performance practitioner was not mine initially. I was drafted into a group at IBM whose mission was to determine if MVS was a viable system with regard to performance. You see, MVS was a bit of an ugly baby when it was born in the mid 70s. I know some will say it never became very handsome but we can debate that later. You had to feed this baby a few more resources than the customer population was expecting. There were a few other troublesome aspects to this baby’s personality but I just worried about the appetite.

A few key design points were not completely met such as the memory capacity. The system required at least 3 megabytes of memory to perform reasonably while its predecessor SVS (single virtual storage) managed quite well with two megabytes. Can you imagine the unhappiness over the requirement for an additional megabyte? Of *course the machine I am recalling was a 370/158 with about 1 MIP of power and the megabyte of memory was very expensive at the time.*

We executed an exhaustive set of tests to determine the performance characteristics of SVS and MVS using several configurations and various workloads. The conclusion of the analysis was that MVS was a viable operating system if improvements were implemented and the appropriate memory configuration was identified. I would like to thank ***John Messenger*** *for sharing his knowledge and insights as he taught me basic performance analysis techniques such as the application of Little's Law.*

My group evaluated the first set of performance enhancements including the SRM "rewrite" as it was called, the point of introduction of domains. A young man named ***Gary King*** *who had participated in the performance analysis of these enhancements for the Poughkeepsie Programming Center explained the new functionality. This began a very significant series of collaborations with Gary over the rest of my career with IBM including the Workload Manager. Gary possesses the one of the finest analytic minds combined with the rare ability to explain complex systems in ways that others can easily understand. The new SRM was improved because it was now more controllable but some of its decisions were quite questionable in my opinion. I soon had my opportunity to apply my opinions and Little's Law to the new improved SRM as a member of the development team. I worked there for several years. I like to call this time in the parameter factory. We could produce parameters like you would not believe. We provided two different means to set dispatching priorities and then added time slicing on top for the advanced user. We added storage isolation, new load balancers, I/O priorities and logical swap controls just to name a few. It was really getting out of hand.*

In 1981 I proposed that the SRM could be significantly less complex using response time as a primary external with internal algorithms that would adjust resource priorities or allocations such as dispatching priority, storage isolation limits etc. based on these simpler goals. The adjustments would be based primarily on profiles of delay as a function of resource allocation (or priority) obtained by state sampling. The proposal defined the vision of the Workload Manager (WLM). The effort had considerable risk and was quite expensive. Although SRM was often cited as an example of MVS complexity, there was no compelling reason to invest; the proposal was put on the shelf for a number of years.

In 1983 I joined a team led by ***Gary Ferdinand****, whose mission was performance analysis and design for the MVS platform and associated products. One of the first challenges was to examine the MVS dispatcher, which was exhibiting a notorious low utilization effect and ominous large system and MP effects. Some of you may remember the so-called BR15 methodology for quantifying this effect, running looping jobs to stop the dispatcher from scanning for work. The driving factor for these effects was the number of address spaces examined by the dispatcher. The dispatcher frequently examined many idle address spaces to find a ready address space. This was exacerbated by MP systems such as the 3084, the first 4 way MP of the S/370 family* *because an event driven preemption implementation increases dispatch rates for MP systems compared to fewer, more powerful CPs with the same aggregate power.*

The requirements identified led to the invention of the "true ready queue" for which IBM was granted a U.S. patent with ***Dr. Ed Cohen****, another great analyst, as co-inventor. The invention allowed insertion and deletion of elements to or from a single threaded queue at any position in the queue without requiring a global spin lock, assuming other appropriate authorization available to the operating system. The S/390 Principles of Operation demonstrates that this is not generally possible using serializing instructions like compare and swap. See CMG 95 late breaking paper "Dispatching Management in MVS from TCBs to Enclaves" published in Aug. 96 Transactions for additional details.*

Although the low utilization and large system effects due to searching ASCBs were eliminated, the introduction of the first 6 way MP, the 3090 600E, demonstrated a significant MP effect that degraded capacity by reducing the MIP rate and increasing the path length or overhead of the dispatcher. This was due to the implementation of event driven preemption resulting in a high rate of interruptions by signal processor (SIGP) to enforce priorities by preemption. The internal throughput ratios (ITR) for the 6 way to 3 way for TSO and IMS workloads were measured at approximately 1.5 or even slightly less with MVS/XA V2 representing poor price performance for an upgrade. You pay twice as much and get 50% more work accomplished.

My solution for this was known simply as "reduced preemption" within IBM, once again demonstrating our crack marketing talent. The solution uses the CPU timer as a preemption mechanism to ensure that low priority work will not run indefinitely while maintaining the ability to preempt on an event basis, as necessary, based on workload characteristics. Sophisticated logic in the SRM manages the timer intervals as well as the status of preemption for individual address spaces. This was described in "MVS/ESA Full vs. Re-duced/Partial Preemption", the 1994 CMG paper by ***Steve Lambourne****. With MVS/ESA (V3), which contained this invention, IBM identified up to 12% ITR improvement for the 600E in IMS environment and ITR ratios above 1.7 in all environments for that processor. Later enhancements in software and hardware improved the ratios to around 1.9. Reduced preemption was so valuable that it was not discussed in the user community and a patent was not pursued; it had the status of a "trade secret" until the Lambourne paper was published. Since Lambourne had published a good deal of information and much more could be obtained by an astute technician from studying trace data, I described the invention and its motivations in the above referenced paper on the MVS dispatcher.*

The MVS storage hierarchy provided many opportunities for analysis, invention and publication particularly with the introduction of expanded storage. The high speed expanded storage provided exceptional responsiveness compared to disk paging and was relatively easy to size for applications with a typical page reference pattern which would allow the least recently referenced pages to be migrated to disk with very modest paging rates. Many numerically intensive programs did not fit the profile; they often referenced storage in a more predictable manner but one that was characterized as skip *sequential. Since the S/390 community was pursuing the advantages of "data in memory", very large matrices were being placed in virtual storage, overflowing central and often expanded storage as well. When insufficient central storage was available, they often "thrashed" in real although targeted to fit in expanded.*

***Catherine Eilert** and I invented "working set management" to address these complicated issues. The invention allowed the SRM to "plot" paging behavior as a function of pages resident in one or more levels of the hierarchy. Productive CPU processing ability at a particular storage allocation was also plotted. This allowed SRM to control working set in central to avoid the damaging effects of pure LRU [Least Recently Used] which reduced the working sets of well behaved applications as it allowed large working sets with no value to large applications. The selection of address spaces to enter the multiprogramming set was also heavily influenced by the knowledge of the characteristics of applications.*

This invention was effective in the numerically intensive environments. UCLA observed at Share that this improvement was one of the most dramatic they had experienced. Ironically it came at a time when other architectures (clustered Unix in particular) were emerging that offered superior price performance than S/390 for numerically intensive workloads.

The working set management effort proved very valuable; it served as the proof of concept of the proposal initially offered in 1981 that SRM controls could be simplified by adding significant additional profiling information and heuristic algorithmic approaches to resource management. The parallel Sysplex initiative provided the business case, dramatically increasing the complexity of the MVS environment. The Workload Manager was a very ambitious project. My primary role in the implementation was in application characteristic profiling and heuristic algorithms collaborating with Gary King and Catherine Eilert. An overview of the WLM approach can be understood from my 1995 CMG paper "The Evolution of the SRM to the Workload Manager in MVS V5" published in the Winter 95 Transactions. A description of the heuristic algorithms can be found in the paper "Adaptive Algorithms for Managing a Distributed Data Processing Workload" published in the IBM Systems Journal Volume 36, number 2 in 1997.

These are very exciting times for all of us in the Information Technologies business. The complexity of the environments that we deal with on a daily basis amazes me. Twenty years ago we came together to talk about very complex entities like MVS or SRM. Today, people working with OS/390 systems can't afford the time to worry about the details that we debated. We expect a great deal from the software and hardware products we use and we are getting much higher quality and more robust solutions. Earlier this year I joined the Consulting business at Candle to lead a team focuses on Parallel Sysplex issues. We had the opportunity to visit the S390 Development Laboratory in Poughkeepsie to get hands-on experience with the technology to refresh our skills. We set up datasharing environments and then tested their availability by pulling coupling links and trashing LPARs. It was a great experience for me. I could appreciate the enormous effort that went into the software systems that were *responding to our tests. Hundreds of thousands of lines of code in the operating system, the data base managers, the communications manager and the transaction managers must work together to provide the robustness promised by the Parallel Sysplex.*

I have an even greater awareness of the complexity of the challenge accepted to develop the Workload Manager. As I said earlier, I proposed the concept of WLM in 1981 and it was considered too expensive. In retrospect, the management that declined the risk was correct; my estimate was about one order of magnitude less than the eventual implementation. The tools and the processes we employed for software development in 1981 were rudimentary compared to those used to develop WLM. That's two big strikes against my idea. We needed to develop the tools, the processes and the people to build function like WLM or datasharing. We needed to grow as an industry. When we had the tools, the processes and the people, we needed to apply them on a small scale. For me personally, the Working Set Management item provided that smaller scale problem to solve before taking on WLM. There were similar examples that led to datasharing.

*As they said in the movies, "if you build it they will come". Maybe they should have added, "and it works as described". In case you have not heard, Parallel Sysplex implementation is happening in a very big way. IBM reports that "over 1500 customers have migrated to Parallel Sysplex. Over half of them are in production doing resource sharing. Customers doing application data sharing grew by 2.7 times in the last year." A strong foundation allows rapid growth. **Bob Maple**, NationsBank states "At one of our NationsBank data centers, we grew from 720 S/390 MIPS in January 1995 to 2030 S/390 MIPS in January 1998. Parallel Sysplex, with its capability to handle multiple OS/390 systems as a single system image, enabled us to handle this significant growth with our original staff of five OS/390 systems programmers."*

*Teamwork is vital in almost every endeavor. I've acknowledged several of the colleagues at IBM that influenced my career. Now I'm privileged to be working with talent like **Steve Samson**, **Ruth Heidel** and **George Dodson** all of whom have been extremely valuable to this career transition to consulting. [Bernie returned to IBM at a later time to continue his work.]*

*I've mentioned two CMG papers that I have used to share my insights with my colleagues. I first attended CMG in 1986 to present a tutorial on SRM.**Tom Beretvas** advised me to get involved in CMG and I thank him for his encouragement. Participation in CMG has been instrumental to my professional growth. I know of no better venue where people hooked on performance and systems management can get together to exchange experiences and knowledge. It is a wonderful thing, and we even have meals and entertainment.*

Finally I'd like to thank my family for their love and support. I thank my mother for bearing me, in multiple senses of the word. I thank my seven brothers for teaching me the spirit of friendly competition. I thank my wife June and my two daughters who have always supported my career.