

From Miami to Orlando – A TPF retrospective

by Waseem Majeed

It is but a short drive from Miami to Orlando, but the trip has been four years in the making and what a remarkable journey it has been. It was from the sunny beaches of Miami, that we touched base the last time. A lot has happened since then; too much in one respect and not enough in another. It will become clear later. In case you missed some of these developments, this is an opportunity for us to retrace our steps and get caught up.

Our first stop on the road to Orlando was in Las Vegas in the spring of 2000. The setting in Miami had been scenic and serene. This time around things were anything but idyllic. The frenetic pace and non-stop activity in the streets was mirrored in the conference rooms and hallways. In Miami there was excitement in the air and the signs of good things to come pervaded the conference. In Vegas, despite the party atmosphere outside, there was purposefulness and a sense of things achieved. There was the realization that not only had the TPF industry weathered the rough times and survived, but that it was undergoing a renaissance of sorts. At the dawn of the new millennium, one got the feeling that TPF had been resuscitated to the point that it had the prospects for providing viable solutions to an ever-widening domain of applications. The Internet, e-Business, e-Commerce and the World Wide Web are the key features of this expanding landscape for TPF. Appropriately, the conference theme was “The New Millennium”. Even the die-hard naysayers have finally conceded that UNIX, or for that matter LINUX, is not going to replace TPF as the high volume transaction processing system of choice any time soon. In fact, distributing a personal copy of LINUX to each attendee as a souvenir would have made that point nicely in addition to being a very practical gift. After all, we do need to be educated in C++, POSIX, UNIX services and a truckload of related tools. If we are to thrive as the next generation of TPF professionals, we have to learn the new vocabulary.

IBM had told the users, in Miami, what they were going to do. In Las Vegas the message was clearly, “we said what we were going to do and we did it”. Len Pellitiere, IBM’s TPF Business Unit Executive, had a very warm welcome for the audience. He provided a good overview of IBM’s direction and recent developments within TPF. Drawing upon the conference logo and location, he worked the gambling theme very nicely into his speech and urged the users not to gamble with their future. I guess his subtle message to users was to bet on a sure thing and to put their money on TPF. He also made several announcements, some of which were particularly significant. He announced the first new TPF license in at least a decade (as far as I can recall). The exciting and interesting thing was that this was new ground for IBM. They were going to partner with OASIS (online payments handling) to develop a new server to handle their transaction business. This server was to be based on TPF for web enabled applications. The other significant announcement was related to the TPF organization. For decades TPF had been part of the Travel and Transportation Industry Division (or Solutions, as everything now seems to be called). Len told the audience that TPF had now been moved under the Software Products Division within IBM. This was an important development and must not have been made lightly. In all probability it came after a lot of internal discussions, reviews and analyses. It represents a consensus within IBM on the importance of TPF and its position at IBM and in the software industry. For any that still doubted this, it is the clearest indication yet, that IBM now views TPF as a very strategic product and an integral part of its future product portfolio.

Stu Waldron, IBM’s TPF Chief Architect, gave a technical overview of recent developments, near term enhancements and the TPF strategic direction. Some future directions of particular interest to the database

community were native SQL support, 32-way loosely coupled support and FARF6. He also announced that the 64-bit architecture POPS (Principle of Operations, for you newbie's) had been signed off. The implication was that someday TPF might be running on 64-bit machines. By publicly mentioning FARF6, IBM was signaling their seriousness and commitment to providing this support.

Especially interesting and worth watching are the vendors and in particular new vendors. To me, new vendors are the sign of a healthy and growing industry. As such, I was somewhat disappointed by the absence of Starpush and Sapiens from the scene. I thought they had some good ideas and some interesting products that could have proved very useful once developed to their full potential. Perhaps, they are still actively pursuing them, but without anyone there to talk about it, we don't have a way of knowing. However, the disappointment was mitigated somewhat, by the appearance of several new vendors (the rookie class of 2000). Two of particular interest were Tachyon, who has an IBM Assembler for the PC, and Dignus, who offer a C compiler for the PC. Both these products should help simplify and ease the constraints associated with mainframe application development. I think they are working with Amadeus. As usual, Thiru Thirupuvanum was back again and as irrepressible as ever. He was busy proselytizing among the crowds and touting the merits of his TPF/GI Tool, which has been available since Dec 97. The slew of features and enhancements included, Source view support for C++ Breakpoints, watch variables, view call stack, seamless integration of TPF and workstation environments. Also added were customizable menus, visual log of events, audit trails and program flow using any standard browser. Statistics with step counts, run counts, view last X count and program include/exclude lists for trace.

Mike Quinn from Sabre applications hosted the "hot topic" on Application performance. This sounded like Déjà vu all over again (sic). At the TPFUG conference in Miami, Galileo had presented a very slick and polished lecture on System Performance Engineering (SPE) playing on the "Mission Impossible" theme. Since then, there has been no further word on how that initiative turned out and Galileo did not speak up during the hot topic. So, it may be that the theme they chose proved all too real. Mike's point was that most performance tools and measurements focus on system metrics and do little to help the applications in designing and developing more optimum code. The collection, reduction and reporting of performance data have a system viewpoint. The results and reporting methods were not relevant or meaningful to an application developer. He asked if anyone had done any studies or developed any tools from an application perspective specifically focusing on typical workloads and profiling of different types of transactions. He wanted to know what is normal I/O, Pool usage, dump rates, frame usage for typical transactions.

There has been a sea change when it comes to TPFDF (Database Facility). It was a rare treat to watch an IBM person, Stu Waldron no less, get up on the podium and speak of TPFDF in such complimentary and glowing terms. He was positively effusive in his praise to the point I was quite taken aback. To appreciate this change one has to know a little bit about the background. Back in the late 70's (Stone Age of TPF), the PARS Revision Project (PRP) group was formed and they began to look at improving the state of the art in the TPF industry. They formed various committees to look at the control program, communications, operations and database developments to see what could be improved in each area. At the same time Swissair, working independently (largely due to the work of the legendary Hans Eisele) had developed a database utility called ACPDB. Since Swissair was part of the PRP group together with IBM, American Airlines (Sabre) and several others, they offered ACPDB as a tool to improve database management in TPF. The PRP group decided to adopt ACPDB for general use provided Swissair committed to providing a list of enhancements within a specified time frame. The Swissair/ PRP agreement expired in 1986 and there was consensus among the ACPDB/TPFDB users that IBM should either adopt TPFDB or provide an alternative.

For whatever reasons, the TPF Lab was reluctant to do this. The stalemate ended when IBM UK, who were marketing and supporting ALCS, saw an opportunity and developed the business case and technical justification for TPFDB. They signed an agreement with Swissair (later Atraxis, and still later EDS-Switzerland) to acquire it and renamed it TPFDF. The TPF Lab was happy with this arrangement until the decision was made to move ALCS under the Lab. As part of this migration of the ALCS development and support organization, the Lab inherited TPFDF.

They never did appear comfortable with TPFDF and were not quite sure what to do with it. That has changed now. There are probably several reasons behind this change in thinking. For starters, despite all its limitations, TPFDF has proved itself a viable and useful tool for database development in TPF. Clearly, it lacks some of the key features of a full fledged DBMS, but it does offer a level of flexibility and functionality that is simply not possible with native TPF code. This is recognition of that. Secondly, while Collection Support is geared towards the future it is not viable yet for industrial strength applications. TPFDF will have to be the workhorse until Collection Support can be further developed or something else comes along. So TPFDF is experiencing a level of development activity not seen since the early 90's when IBM UK first acquired it from Swissair. Takao Inouye (IBM) gave an update on TPFDF and stated that PUT13 would be generally available on April 28, 2000.

In spite of these developments, it still seems like the Database area is the 'final frontier' for TPF. The industry owns some of the largest databases in commercial use, yet the state of the art is still essentially 1970's technology with some pockets (pun intended) of excellence. There is TPFDF to be sure, and we have pockets and folders. Then there is Collections Support and the UNIX file system each of which can handle a certain class of applications. We also have TPFAR (Application Requestor), the TPF client side middleware, which conforms to IBM's DRDA (Distributed Relational Data Architecture). Even though this is proprietary to IBM and the database resides in a DB2 system, it does use industry standard SQL and allows TPF to extend the range and class of applications it can support. Some software vendors have signaled their intent to develop products conforming to DRDA but it is not clear which, if any have actually done so. Despite this, the fact remains that a fully functional and capable DBMS is still lacking. While TPFDF has its limitations, it does have one big advantage. It supports a common API between C, Assembler and PL/TPF (Sabretalk to us old old-timers) and provides standard data structures that can be used to handle a variety of database situations. Thus, it can be a good bridge between legacy TPF data structures and the databases of the future.

While TPFDF does provide valuable application services, there are limitations to what it can do. Considerable work needs to be done to improve performance and functionality. The good news is that IBM seems to have caught on and is willing to invest in TPFDF. While they continue to improve the new tools and look forward to the gradual adoption of the new technologies, they are now accelerating the pace of TPFDF development. The objective is to improve what is already available and working well for many applications. However, users need to work closely with IBM to define requirements and guide future directions. One other thing in our favor was that Takao was quite bullish on TPFDF. He appeared keen to play a proactive role in its development and support and was ready and willing to work with users within a diverse customer base.

I posed the question, to the audience and to IBM, if they had considered providing some compatibility and co-existence between TPFDF and Collection Support (CS). It should be possible to define a collection as a TPFDF type with a set of methods that map to the existing API. Conversely, it should be possible to

manipulate a collection using the TPFDF API. If CS is the future, then compatibility with existing structures needs to be a requirement. The CS API should allow existing TPFDF databases to be supported. For performance reasons, it may be necessary to define existing files as collections, but have DF process them under the covers. The situation is not so different from when TPFDF was first being widely used. In some instances customers chose to set up the database structure as a DF file, but continued to use FIND/FILE and native TPF code to process the data to overcome some performance and functional limitations in TPFDF. A final thought on this. IBM and the TPF industry need to have a database strategy. Right now it seems this is the least defined area within TPF. We have to better understand the options and choices and take the steps to conquer the last “undiscovered country” that the database area represents.

From a customer perspective though, attending this conference was a very sobering experience. The joy and excitement of witnessing all these new developments and TPF trends was tempered by the stark realization that we, as an industry, have lagged behind. While other industries have moved rapidly to adopt the new technologies and adapt to the changing realities, we (the TPF user community) seem to have fallen behind the eight ball. Our development tools and methods are woefully inadequate to meet the challenges of the new millennium with its new TPF and new business models and new opportunities. We are behind in our education and skills development. Our operating environments are not state of the art and we have been slow in moving to the new standards. This conference showed clearly and unequivocally just how far behind we have fallen not only compared to other industries but also with respect to what TPF now has to offer. Too much has happened on the TPF product side and not enough on putting the new functionality to work. Of course, there are exceptions and some talented teams at various customer installations are working and experimenting with the early adoption of the new tools. However, by and large, one was reminded of Dickens line from a Tale of Two Cities, “It was the best of times and it was the worst of times”.

Our challenge is to find the best ways to take advantage of all the new tools and facilities that TPF offers, in designing solutions for our businesses and customers. We need to understand the advantages and disadvantages of each tool and to figure out which solution or approach provides the best result in any given situation. In summary, there was confidence and reassurance that TPF was alive and well and by all indications looked set for another period of growth and development. With growth comes opportunity. The rewards belong to those who seize the opportunities. It is not enough to do the right thing or to do things at the right time. One must do the right thing at the right time. TPF was clearly on the cusp of something big about to happen.

The next stop on our TPF journey takes us to Tucson and another set of surprises. I arrived in Tucson fully expecting to be greeted by the typical parched desert with dust devils, swirling winds, searing heat and barren landscapes. What a surprise to find a veritable oasis. You wouldn't call it lush, but there was plenty of vegetation. To be sure, not a blade of grass was to be seen, but there were plenty of other types of greenery. The architecture seemed to blend in with the landscape. Eons ago there must have been a seabed or river valleys here. And the wind was remarkable for its calmness almost to the point of total absence. The clouds seemed to form and just hang still in the sky for the longest time till they eventually burned away. The cool breeze coming down from the mountains made for pleasant mornings and relaxing evenings. It certainly did not fit my expectation of what a desert is. Anyway, we did not come for that and the conference atmosphere was very businesslike and that is where IBM exceeded my expectations.

Just when you think things can't get any better, they do. If Miami had been the grand invitation with IBM promising this and that and everything in-between, then Vegas was the appetizer. We were tempted with

morsels of succulent stuff with tantalizing hints of more delights to come. If the analogy holds, then Tucson was certainly the banquet. What a grand feast it was, as IBM served up enhancement after enhancement and features galore. There was something for everyone from Comms, to Systems to Database and the entire gamut of TPF products and services. I am not one to be easily impressed, or readily intimidated by technical jargon. But I have to say, I was truly amazed at what IBM had been able to accomplish and deliver. My head reeled from the effort of trying to absorb the technical details and information on the dizzying alphabet soup of products and services.

The mainstreaming of TPF was almost complete. The retooling that started at the beginning of the decade with development of TPF 4.1 was in its final stages. The new starship TPF had arrived with the new millennium and was ready to head for the galaxies and new worlds to explore. For over three decades TPF had been driven by the industries it served. On the one hand airlines like American, United and Swissair had been pushing the envelope on speed, response time and ever-increasing database sizes and functionality. At the other end of the spectrum we have had the banks and other financial institutions working for greater integrity, reliability, availability and fault tolerance. These two effects have now converged quite effectively in the new TPF. In an interesting twist, it is the technology that has taken over and is pulling TPF head long into the future. It is no longer the industries or the customers that are pushing TPF but rather technology and product innovation that are pulling TPF forward. Mired in financial difficulties and preoccupied with the consolidation and retrenchment in their own industries, the customers have been struggling just to keep up, let alone harness the new power and energy of TPF. Propelled by the new technology TPF appears to have left its traditional power base in search of new industries and new users. Outside of its protective niche, TPF has found the going to be tough as it struggles to compete with the established players in the Internet services market. But TPF has a big break. We have to travel to Orlando to learn more about that.

I was not in Toronto, so have to fast forward to Orlando for our final stop on this Journey. In the land of Mickey, things were anything but Goofy. Once again, Len was center stage and had some important messages for TPF users. Post 9/11, the industry is facing difficult financial times and confronting some key technological challenges related to changing business models, scalability issues and competitive pressures from alternative platforms and negative coverage in the trade press. His message was that TPF is still the leader and users must assess and evaluate any plans to migrate to competing platforms to ensure they are getting a good return on their investment. For its part, Len assured the audience that IBM would continue to invest heavily in TPF development for the future of the product and their user base. As part of that commitment, Len announced IBM's statement of strategic direction to port TPF to the new 64-bit z/Architecture. There will be a TPF 5.1 in our future

Another key statement by Len was that TPF is still the leader and optimal solution for high-end OLTP requirements with unmatched availability, reliability and price/performance characteristics with an installed user base running 1.5 to 2 billion Transactions per day in production systems. Recent upgrades to support 32-way loosely coupled processors, each with up to 16 I-streams make it possible to configure a system with 512 processors and a capacity approaching 100,000 MIPS. With the implementation of FARF6 a TPF system can support an addressing capacity of 72 quadrillion. I forget how many stadiums full of apples or oranges that is or if they were laid end to end what galaxy they would reach.

Len stated that some users are offloading portions of the application value chain from TPF to relational Databases on alternative platforms and realizing costs measured in dollars-per-transaction compared to TPF at 6-10 cents average for the airline industry and less than a penny for credit card authorizations. IBM's point, if there was a problem; it was always the application and not the platform. Before investing in expensive alternatives, users should consider TPF centric solutions e.g. TPF access to relational Databases on OS/390 via the OSA Express card.

The trade press reports comparing TPF to other platforms are often inaccurate, misinformed, unproven and subjective. They don't have the facts so they end up comparing apples to oranges. IBM will need to do a better job in providing accurate information to counter this type of negative publicity. Customers can help out by showcasing their products and success stories. As part of that strategy IBM hired a topnotch consultant from ITG, Brian Jeffreys, to compare high-end OLTP systems. His conclusion, TPF rules across a wide spectrum of industries and application profiles. As reassuring as this result is, one still has to wonder why all the online booking engines/systems created over the last 5 years are non-TPF. Just as disconcerting is the fact that few, if any, are based on IBM hardware, software or databases. What do these CTOs and CIOs know that IBM is not able to see or tell us?

The other significant event was the presentation of the ITG consultant's report on TPF compared to the best of the rest in the OLTP arena. This consultant, a former economist, was featured as the main theme presenter. Apparently he had been hired about a year ago and was going to present his findings and conclusions at this conference. He compared TPF against competing technologies on several factors with different configurations of hardware and across different industries (e.g. retail, banking, credit card authorization etc). Not surprisingly he ranked TPF at the top in almost every category. What was surprising is that he singled out Himalaya for special mention. He did not view it as a viable replacement for TPF under present circumstances and was not sanguine about the prospects for the Itanium chip changing that. It would be wise for all TPF users to review his report and at the very least pause and reflect carefully on the technical and financial merits of any strategic platform choices they are contemplating.

IBM announced a statement of strategic direction to port TPF to the 64-bit z/Architecture to address challenges of high volume and large address space programming models common to other platforms. The strategy includes adopting a suite of tools to support standard application development using C/C++ now and JAVA in the future, for operating system services, networks and web and email servers. The intent is to port TPF to the new 64-bit z/Architecture family of processors. So, TPF 4.1, code-named NBR (Next Big Release) but sometimes referred to as LBR (Last Big Release by TPF detractors), turned out not to be the end of an era, but the beginning of a new and even more spectacular chapter in the storied history of TPF.

Another very interesting announcement (or should I say a project of interest) by IBM was that WORLDSPAN would be using TPF hosted E-mail for up to 30,000 travel agent accounts across USA and Europe. IBM has been steadily building up the Web/Internet capabilities of TPF over the last few years. Except for a few isolated and small-scale operations they have had little to show for their efforts. This could be a major coup for IBM and set the stage for more web-related applications on TPF

Once again, the Database subcommittee was of special interest to me. Mike Shershin provided a very good overview of FARF6. Dan Jacobs presented an updated on TPFDF PUT16, which was which was

announced for general availability on Oct 16, 2001 and PUT 17 which was not yet generally available, as of yet. He also talked about some enhancements in the works. Fred presented the EDS-Switzerland (formerly Atraxis and ex-Swissair) experience with cutting over to TPFDF PUT15, the key element of which was the Common Entry Point (CEP) project. He listed the main reasons for the development of CEP, reviewed the essential changes and restructuring of the middle-ware code, and went over the pre- and post-cutover activities related to the implementation of CEP.

Earlier we briefly reviewed the historical background of TPFDF, so let's focus for a moment on the technological aspects and current state of affairs. First developed as a file utility in the late 70's, TPFDF is once again on the front page of IBM's development plan after a period of slow growth in the 90's. As the workhorse of TPF database development at several large installations, TPFDF was somewhat sidelined and over-shadowed by the development of the new C and Object based API's. There are several factors driving this change in fortunes. For one, the re-invention and re-emergence (or should one say survival) of the mainframe as the centerpiece of the enterprise and global distribution computing models, has benefited mainframe tools in general. For another, the lack of viable and proven alternatives has meant continued use and reliance on the traditional tools. The new products like Collections, while interesting and attractive in many ways and more compatible with future languages and application API's have been slow to gain acceptance. They have yet to prove themselves in industrial strength applications. Until that happens, we can expect TPFDF to carry the load. More importantly, it is the main TPF product providing a standardized and structured organization for database storage and retrieval with a common API that can be used by all applications. So, it provides an important bridge to the future. A bridge is a structure that takes us from here to there. We know exactly what the 'here' is, but unfortunately, no one has quite built a description or articulated a vision for where, the 'there' is and what it looks like and how far it is. That is the key challenge for the TPF database user community.

Designed and developed over two decades ago, and loosely based on the hierarchical paradigm of IMS (IBM's main commercial database prior to the development of DB2), TPFDF has lately been showing its age and need for serious upgrades. The traditional TPF computing model, based on short simple I/O driven transactions, that TPFDF was designed for, has long been outdated. Applications need to access much more data, the business logic is more complex and functions are more tightly integrated than ever. The demands of more complex but still high performance functions have stretched its fragile capabilities to the limit. No one has quite pushed the envelope as the Sabre system. From very high volumes to unprecedented levels of concurrency with ever greater complexity, TPFDF has been sorely tested and found wanting in several areas. There are three critical success factors for the TPF database community:

The first is the database software. Strengthening, reinforcing and beefing up the physical infrastructure of the existing database middle-ware is one option. I think this need was recognized by IBM and is being addressed. The Common Entry Point was a key aspect of this restructuring and reinforcement and provides a solid foundation for DECB, FARF6, LLR (Large Logical Record) support and other infrastructure changes and enhancements in the TPFDF code. An alternative would be to build something completely new or port something from the UNIX or MVS worlds.

A second key component for success is functionality. Since its inception over 20 years ago, the API has remained largely static with few enhancements. There are several inherent limitations in the TPFDF paradigm. The record at time (RAT as it's called) processing needs to be supplemented with more set-

oriented operations, such as the DBMOD, which can update large sets of data. There are many classic linked list applications (such as Queues), which could benefit from list-oriented operations, which DF does not provide. So, users have to rig their own solutions to support the highly concurrent activity required. TPFDF does not provide enough flexibility and controls to support different granularities of HOLD (e.g. file, sub-file, and record level) nor does it have the different modes of LOCKS (e.g. protected read) that are sometimes necessary for concurrency and integrity. These are just a few examples to illustrate the point. The paradox is that Collections has rich set of list-oriented functions but cannot sustain the performance or concurrency required by a typical DF application. TPFDF has the muscle but not the intelligence. Somehow we have to resolve this conundrum and find a way to get the best of both worlds. Perhaps the new C++ API's IBM is working on can bridge the gap. Enhancing TPFDF with a C++ layer is one option. Other alternatives are Object DBMS or SQL. Native SQL support on TPF is something IBM has talked about for years. Despite what some people might believe or think, SQL is here to stay and will be around for a long time. So, perhaps it is time for IBM to dust off their plans and take a serious look at SQL.

The third leg of the database tripod is Education and training. We tend to use the tools before we have mastered the techniques. The tool then becomes a crutch and an impediment to learning the underlying techniques and technology. Do we know how to design large complex databases? Do we know how to program complex database operations to support concurrent access while preventing deadlocks and maintaining integrity? Do we have proven well-thought out and rational methodologies to develop database applications? Do we just cobble something together based on instincts and gut feeling and hope it works?

IBM is fond of saying, "it's the application, not the platform". To that I would add, "it's not only the application or the platform, it's the database too (not necessarily DB2)". Just like a house is only as strong as the foundation it sits on, an application is only as good as the database it is built upon. Right now, the database could be the weakest link in the value-added services provided by the communications, applications, database and operating systems layers in processing a transaction. Several new players, such as COMPAQ, Oracle and others have entered the playing field. Armed with some sophisticated database technology and bringing with them a rich legacy of experience in the database arts, they pose a potent and powerful challenge to TPF. Despite what IBM might think, these guys are nimble and quick, they learn fast, adapt well, and time is on their side. Unless we address the limitations in the database area, forge a vision of what the future database applications will look like and create a roadmap to get us there, TPF cannot hope to compete effectively in the long run. It is only a matter of time, before we hear the words, "You are the weakest link, bye-bye" and get booted out of the game. Before that happens we have to morph the database from being "Conan The Barbarian" into one more like "Spiderman". Times have changed, business models have changed, economies of scale have changed, and the world has changed. It is time for us to change and create a new database universe. Can TPFDF get us there, or will it be 'a bridge too far'?

So what do all these developments portend? It appears that IBM is under some pressure to produce tangible gains from the heavy investments they have poured into TPF in the last few years, especially in the area of Web services. The outward looking, expansionist strategy, geared towards adding new users and new industries to the installed TPF base, has not panned out as IBM had hoped. The announcements at Las Vegas about the partnerships with OASIS and another small company to use TPF as a web server have not resulted in anything consequential. So, IBM, at least for the present, appears to

have turned their attention inward. They appear to be focusing on expanding the range of installed applications within the existing TPF customer base. They would like to bring more web-oriented services to the TPF fold and grow the business that way.

The WORLDSPAN E-mail deal appears to be part of this new strategy. It will be the first major and significant test for TPF's web related services both from a technical viability and financial feasibility aspects. If this is a success, it could open the floodgates. It would not only bring a host of new applications to existing TPF customers, but could eventually allow TPF to crack the Web and Internet services market. The potential market is huge and TPF stands to gain a sizeable piece of that. This is definitely one to watch and is a very encouraging piece of news. However, we shouldn't lose sight of the fact that while Web services are hot and the newest "in" thing and the latest trend, the pendulum will eventually swing back the other way. These days, every problem seems to require and find a solution in Web services. A lot of development and marketing activity is geared towards meeting these needs. Once the current web services wave has peaked, there will be renewed interest and development activity in the more traditional areas a few years down the road. IBM also needs to focus beyond the current web priorities. .

The second aspect of the inward focused strategy is to retain the existing market share and installed customer base by preventing and heading off defections. For years IBM ruled the roost of their niche market and lived off the fat. Defections were never a serious challenge and were typically shrugged off as the user's loss. With the decline of the mainframe, consolidations within the TPF industries both travel and financial, the emergence of the mega-carriers and the Global Distribution Systems that support them, they are confronted with a changed landscape. The recent events of last September, the downturn in the economy and availability of viable alternatives to TPF have added to IBM's woes. They have to fight for every slice and piece of their customer base to maintain their market share. IBM brought out their heavy guns and fired the first salvo in what should be an interesting battle for the high-end OLTP market. Armed with facts and figures, they made strong and forceful statements about the strengths of TPF and its advantages over the competition. To reinforce their point, they hired the ITG group to conduct a study and present their findings at the TUG. They questioned the wisdom of making huge investments to realize small savings in what they call 'the look to book' slice of the reservations applications. This new dual strategy is likely to pay off and should eventually allow TPF to expand into new markets.

Even though the mood was somewhat subdued compared to the last few meetings, nevertheless IBM sounded upbeat on the prospects for continued strength in the TPF sector. The resurgence of the mainframe and IBM's investments in the core TPF services should begin to pay off. However, there are still problems to overcome. During the 90's the mantra had been "the mainframe is dead" and "TPF is fading away". The pundits were wrong on both counts. Not only have the mainframes survived, but also appear to be set to make a big comeback due to a host of factors (addressed in Brian Jeffrey's report). The old mantra has been replaced by a new chant, "Yes, TPF will be around, but no one will develop applications on TPF" or "Business logic does not belong in TPF". Where's the logic in that kind of thinking? The challenge for IBM is to overcome this mindset. While it is true that not every application belongs in TPF and some functions are best distributed, the pell-mell rush to replace TPF with UNIX or its clones is somewhat misguided.

There is no "one size fits all solution" and there is still a significant amount of work that could be done in TPF. Could this also be part of the problem for TPF? Is IBM trying to do too much in trying to make TPF

all things to all people? Perhaps a more restricted and better-focused product strategy leveraging the traditional strengths of TPF might be more successful. Are there TPF services/modules that could be moved over to UNIX/LINUX or even OS/390? Could TPF actually benefit more, if IBM invested in developing applications/systems that can co-exist with TPF? TPFAR is a good example of this approach and, from a purely technical viewpoint, has been one of the best products developed by the TPF Lab in recent years (mho). Even though it has been quite successful and effective in the limited cases where it has been deployed, it is not clear why its usage has not been more widespread. Perhaps IBM is partly responsible for this. After the hoopla surrounding the initial launch of this product, IBM has been somewhat muted and silent on TPFAR at the last several TPFUG conferences. The result is that it has not yet lived up to its hype or real potential.

So what is the future for TPF and where does it need to go from here? Some of the answers rest with IBM. Their investment decisions, strategic choices, development directions, tactical plans and marketing savvy will be the factors that set the compass heading for TPF. Users, customers, developers, 3rd party vendors, technical leaders, IT managers and most importantly CIOs and CTOs will determine where it actually ends up. Do all the new distributed and web-based tools services, products and technologies and their related architectures, mean the end of the monolithic centralized applications and systems where TPF rules? Probably not, they have a continuing raison d'être and will remain relevant for the foreseeable future. While we do need to capitalize on the emerging trends and take advantage of the new technologies and tools, we should not completely abandon existing frameworks. (Let's not throw out the baby with the bathwater!) However, IBM has a tough sell ahead of it. IBM has to convince users that with the new development tools, standard application interfaces, myriad of connectivity options, there is no reason not to develop on TPF. If there is technical justification, makes financial sense and is based on sound business decisions, then why not? One can argue that with the recent enhancements to TPF, the development environment is as rich as that on any other platform. I hope users have a better plan for TPF 5.1 and don't respond in the same way as that IBM engineer. When told about the development of the microchip, he asked, "But what is it good for?" Rumor has it that he is now working as a pizza delivery guy somewhere in Smallville. Think bigger, think different, think well, and most of all rethink TPF.

The comments, remarks, and opinions expressed in this article are those of the author alone, and are not necessarily shared by any other person or business.