

# The Value of Using Name-Value Pairs

Operations and Coverage Subcommittee

---

Jesus Galvez  
z/TPF Development

# Disclaimer

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

# What is Name-Value Pair and Name-Value Pair Collection?

# What is a Name-Value Pair?

- Makes data available anywhere in the application without modifying the application linkage. For example, NVPs can be used for name and password credentials by issuing a set and get API with the name (similar to that of a hash map).
- Name-value pairs can also be used to annotate messages processed for the benefit of name-value pair collection. For example:

```
MsgType=Shopping  
Origin=Mobile  
Channel=Partner  
Customer=MonsterTrvlSite
```

# What is Name-Value Pair Collection?

- Name-value pair collection makes use of name-value pairs and ECB owner names to identify:
  1. Messages processed by the system.
  2. Average resources used to process different messages.
  3. Code packages or phases of processing used to process messages.
  4. Average resources used by the code packages or phases of processing.
- Name-value pair collection leverages the same counters as ZMOWN and the ECB Resource Monitor.

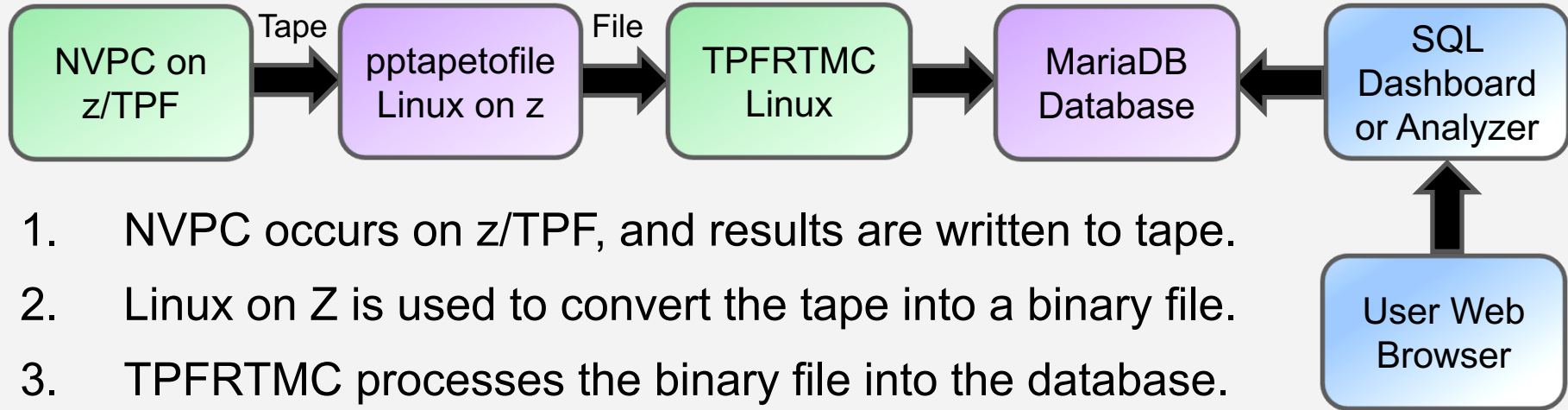
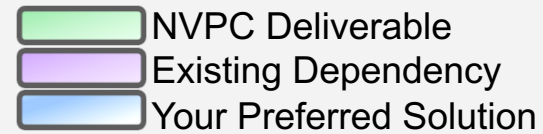
# Value Statements

- A coverage programmer can use name-value pair collection to gain new insights into system resource usage and identify the source of problems in as little as 1/20<sup>th</sup> of the time previously required.
- A capacity planner can use name-value pair collection to determine additional physical assets required given new resource usage metrics and the message growth rate of a specific service.
- An application architect can use name-value pair collection to identify inefficient code packages that can be refactored to improve system performance.

# Name-Value Pair Collection in the big picture

- Use Data Collection/Reduction, Continuous Data Collection (CDC) and other tools to understand system resource usage across the system.
- Use Owner Name Collection, Software Profiler (ZTRAP) and other tools to understand resource usage in broad strokes or for particular code areas.
- Use **Name-Value Pair Collection** to “zoom in” and understand resource usage at:
  1. The message level.
  2. The code packages or phases that implement the message processing.

# Name-Value Pair Collection Architecture



1. NVPC occurs on z/TPF, and results are written to tape.
2. Linux on Z is used to convert the tape into a binary file.
3. TPFRTMC processes the binary file into the database.
4. Use and SQL dashboard or analysis package of your choice to create questions and view results in the form of trends. We have demonstrated possibilities with the open source package Metabase.



# Name-Value Pair Collection Status

- PJ44321 – 4Q2017: Online NVPC facility. For more information, see. Name-Value Pair Collection  
[https://www.ibm.com/support/knowledgecenter/SSB23S\\_1.1.0.14/gtps3/cnvpcp.html](https://www.ibm.com/support/knowledgecenter/SSB23S_1.1.0.14/gtps3/cnvpcp.html)  
ZCNVP command  
[https://www.ibm.com/support/knowledgecenter/en/SSB23S\\_1.1.0.14/gtpo1/cmd\\_zcnvp.html](https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.14/gtpo1/cmd_zcnvp.html)
- PJ45264 – 2Q2018: Offline TPFRTMC facility.  
Testing continues. We plan to provide sample SQL and videos of how we set up and use Metabase to create questions and view trends.
- Name-value pair exit logging sample code. Helps you identify where you have not set NVPs.  
[https://www.ibm.com/developerworks/community/blogs/zTPF/entry/Name\\_value\\_pair\\_exit\\_logging\\_sample\\_code?lang=en](https://www.ibm.com/developerworks/community/blogs/zTPF/entry/Name_value_pair_exit_logging_sample_code?lang=en)

# Name-Value Pair Collection Performance

# Name-Value Pair Collection Performance Disclaimers

- Your results may vary depending on a wide range of factors (application path length, number of owner name changes, skip factors, etc).
- All results are compared to baselines with no collection running.
- Testing was performed with a short application path length. Typical customer applications have an application path length of 3 to 5 times larger which will further dwarf the impact of NVPC.
  - Drove roughly 4k messages per second.
  - LPAR with dedicated I-streams.
  - Using variety of name-value pairs and name-value pair count (0 to 14).
  - Using Message Skip factor from 0 to 9999 and Owner Name Change Skip factor of 0-9.

# Name-Value Pair Collection Performance Results

Message Skip Factor	Throughput Degradation	Samples Lost	Samples Captured
0	7-8%	0.2%	~6000/sec
1	3-4%	0.2%	~3000/sec
10	0-1%	0.1%	~500/sec
20	0%	0	~250/sec
9999	0%	0	~0.6/sec

- Owner Name skip factors had negligible impact on results.\*
- Setting NVPs has virtually no impact (NVP set API has been optimized).
- The number of NVPs collected had no discernable impact.

\* Highly dependent upon the number of owner name changes and the owner name skip factor.

# Name-Value Pair Collection Performance Statement

- With well chosen skip factors, NVPC has negligible performance impact on the system.
- Ensure the skip factors are large enough that you don't lose samples by collecting faster than the tape writes can occur. Losing samples may skew the calculation results in TPFRTMC.
- Choose skip factors such that the performance impact to the system is negligible.
- If more samples are required, run the collection for a longer duration.

# Name-Value Pair Collection Performance Guidance

- As a general guideline, ensure that you have around 1000 samples for each of your most common messages to have a reasonably tight 90% confidence interval in the majority of your results.
- If a particular type of message has high variability, you may want to increase the number of samples for that message type to 10,000.
- Suppose booking messages make up 1% of our total message mix and we want to ensure about a 90% confidence interval: then we would need about 1,000 booking samples and about 100,000 samples total. Given a rate of 2,000 system messages per second with the message skip factor set to 20 (100 messages collected per second), we would need to run name-value pair collection for about 17 minutes.

# Name-Value Pair Collection Ease of Use

Why use Tape?!

# NVPC Ease of Use

## Why did we use tape?

Minimal production system impact during collection.

## Then how can it **POSSIBLY** be easy to use?

In playbacks, we demonstrated how to automate the entire process.

1. TOS Raven scripts automatically:
  - Starts and stops NVPC. Notifies Linux on Z tape is ready for processing.
2. A CRON job on Linux on Z automatically:
  - Mounts and processes the tape to create a binary file. FTPs the binary file to a directory on the Linux x86 box.
3. TPFRTMC on the Linux on x86 box automatically:
  - Processes binary file and stores the results in MariaDB database.



# What's Next?

# We're not done with Name-Value Pairs, 2018 and beyond...

## More Accurate

- New NVP APIs to accommodate various application architectures. First new APIs will be the get and set NVP list APIs PJ45192, planned for 2Q2018.
- NVP support in Java.

## Easier to use

- Automatically pass NVPs to REST, Java, business events and etc.
- Automatically set NVPs by configuration for REST, Java and other interfaces.
- Formal documentation of how to use NVPs and all related APIs.

## More Value

- User defined NVPC metrics such as proprietary APIs, business metrics, or etc.
- NVPC in real time.
- Capture diagnostic information based upon specific NVPs in an ECB.
- Predictive Analytics to increase IT and business effectiveness.

# Plug

Wednesday education session with Josh Wisniewski to revisit how to use:

- Various NVP APIs.
- ECB Owner Names.
- NVP usage in various application models.

# Summary

# Summary

- NVPs are easy to set and performant. Eventually, they will be automatically set by configuration in some circumstances. The system manages NVPs in many circumstances.
- Efforts are underway to help manage NVPs more easily for your unique programming models.
- Owner names can now be set automatically by configuration (See PJ44680 – Program Configuration File).
- NVPC can be run without impacting the load on your system, and is easy to use as it can be executed end to end without intervention.
- You customize how you view and analyze NVPC results.

# The Value of Using Name-Value Pairs

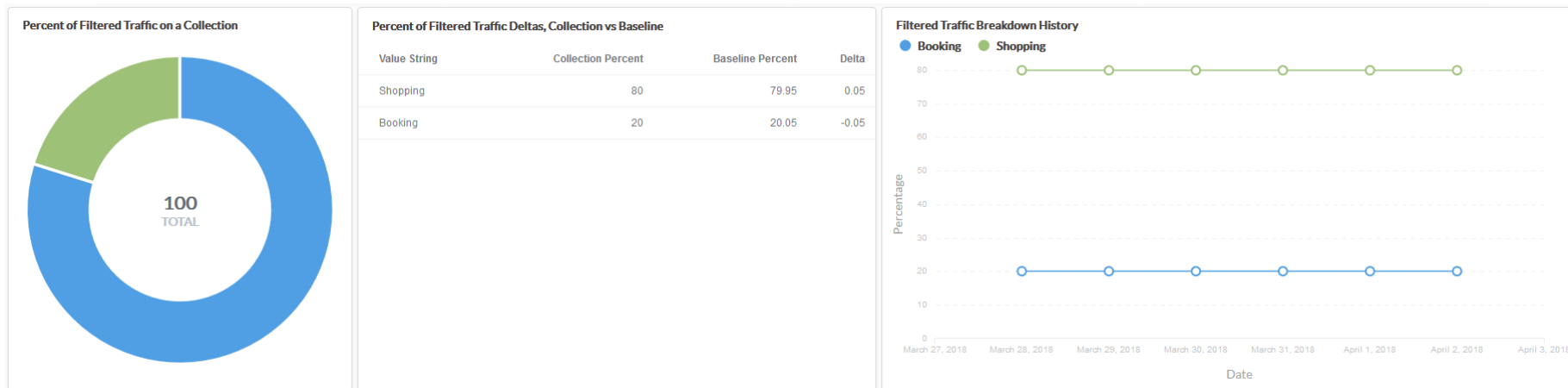
# The Value of Using Name-Value Pairs

- With a minimal investment, you can realize a tremendous amount of value from NVPs.
- In the following charts we will:
  - demonstrate the value of setting a single NVP.
  - demonstrate the value of setting a single level of ECB Owner name.
  - demonstrate the additional value of setting a single NVP and a single level of ECB Owner name together.
- The following charts show the open source package Metabase as our example implementing the wire frames we showed last year. However, you can use your preferred SQL package.

# The Value of Using NVPs

Suppose our application has a single name (“MsgType”) and the possible values: (“Shopping”, “Booking”, “Ticketing”, etc) to describes the type of message or utility that is executing.

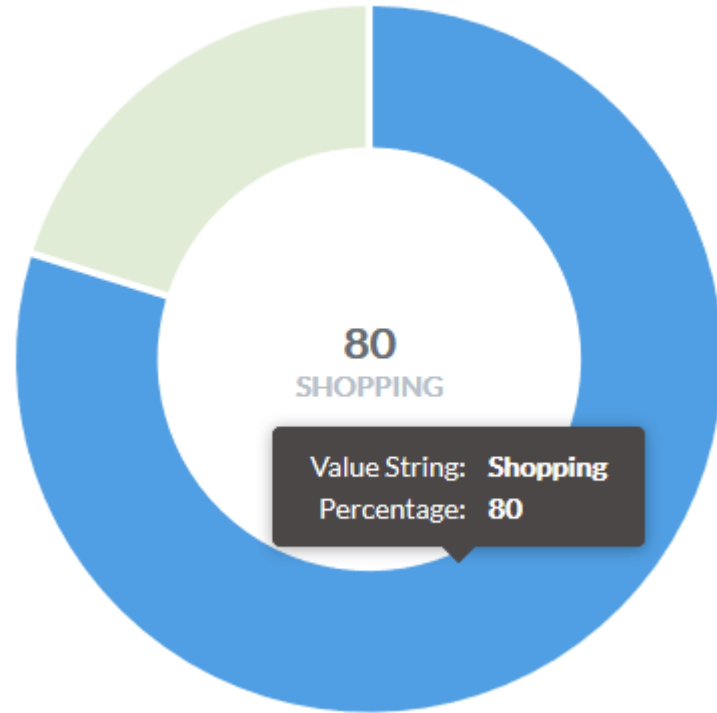
By running NVPC we are able to capture data points to answer all sorts of questions:





# Average message mix for a collection

Percent of Filtered Traffic on a Collection



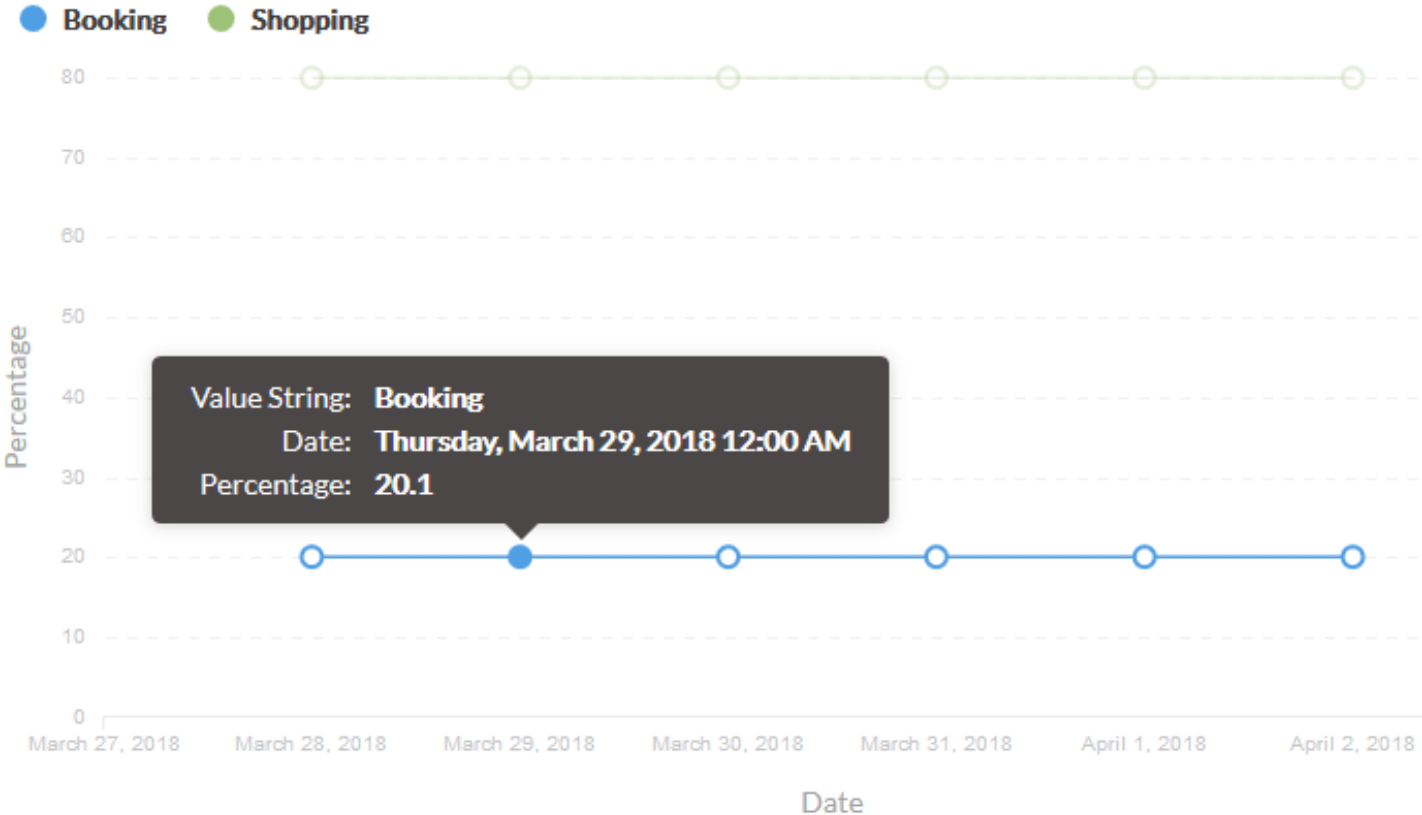
# Average message mix for a collection compared to baseline

## Percent of Filtered Traffic Deltas, Collection vs Baseline

Value String	Collection Percent	Baseline Percent	Delta
Shopping	80	79.95	0.05
Booking	20	20.05	-0.05

# Average message mix historical trends

## Filtered Traffic Breakdown History



# Average resource usage per message compared to baseline

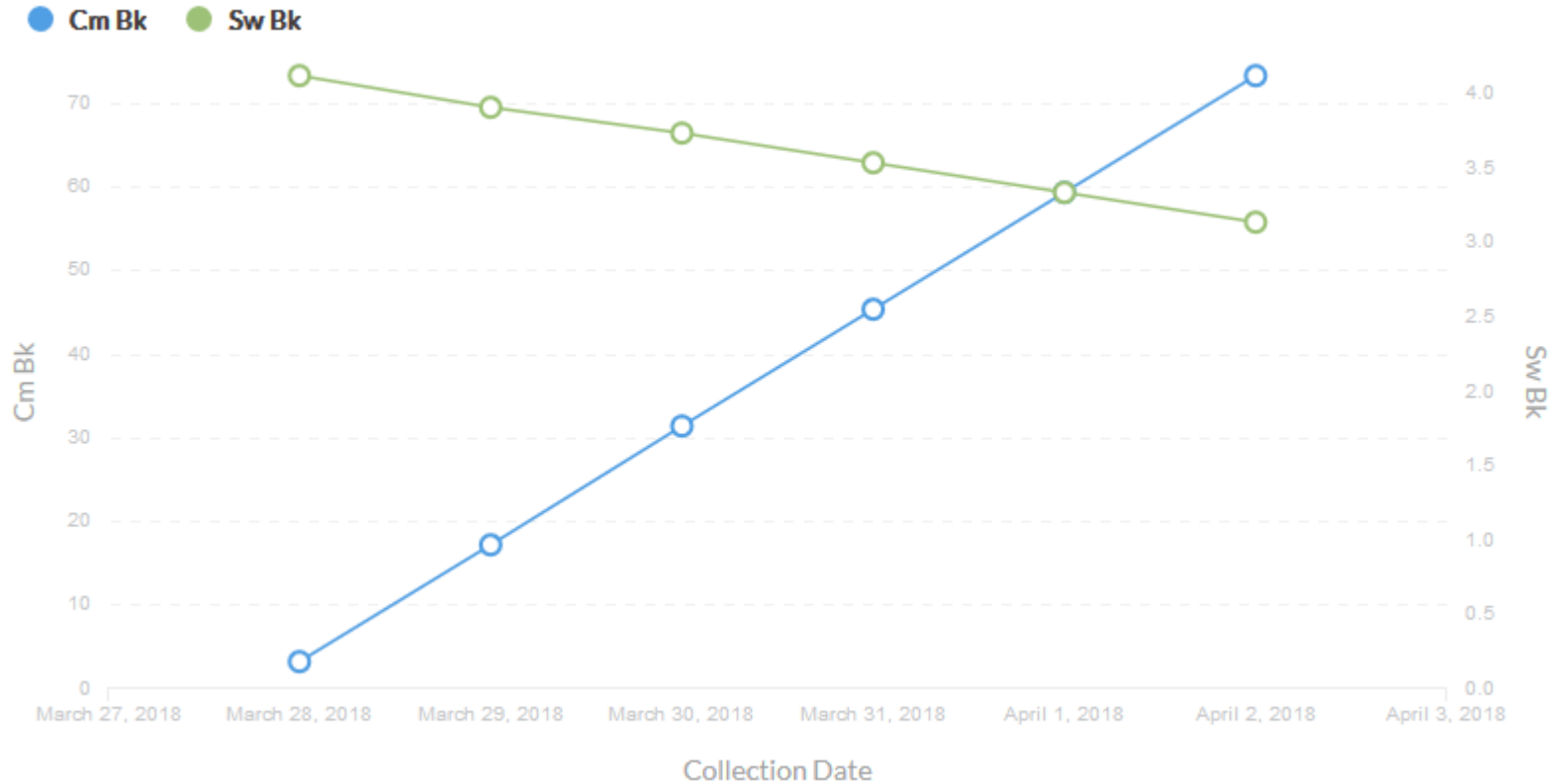
## Filtered Metrics Deltas

Metric	Current Value	Baseline Value	Delta
cmbk	73.3	31.32	41.98
swbk	3.14	3.72	-0.58

Metrics are the same as ZMOWN and ECB Resource monitor:  
Existence Time, 1MB Frames, COW, IOs, TPFDF, ...

# Average resource usage per message history

## Filtered Metrics History



# Average resource usage for a type of message vs. baseline

Horizontal NVP Filter

msgtype=booking

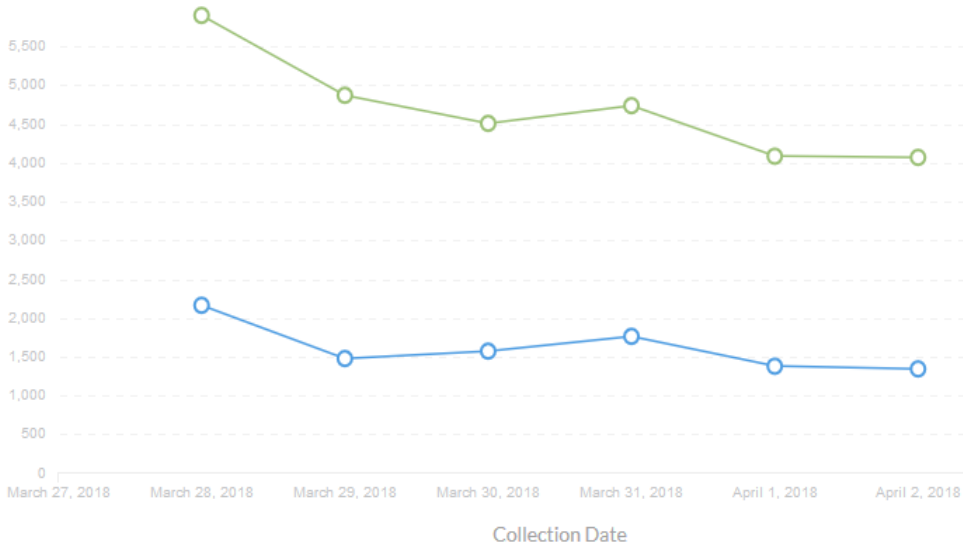


## Filtered Metrics Deltas

Metric	Current Value	Baseline Value	Delta
cpu_used	1,341.97	1,678.89	-336.92
exist_time	4,070.51	4,824.19	-753.68

## Filtered Metrics History

Cpu Used Exist Time



All of those data points from  
just one NVP!

# The Value of Using Name-Value Pairs

- Suppose we set ECB Owner Name Hi for the code package by configuration (NO code change). It can have the possible values: COMMS, PARSE, FRAUD, SHOP, BOOK, FINALIZE, etc.
- Just like the MsgType NVP, NVPC will be able to show you:
  - Average percent of messages that use this code package compared to baseline. And the historical trends.
  - Average resource usage by a particular code package. And the historical trends.



# The Value of Using Name-Value Pairs

- Lastly, you can use the NVP MsgType and ECB Owner Name Hi for the code package together.
- Just like the MsgType NVP, NVPC will be able to show you:
  - Average percent of MsgType messages that use a particular code package. And the historical trends.
  - Average percent of MsgType messages that use each code package ever called by that message type. And the historical trends.
  - Average resource usage by a particular code package by a specified MsgType. And the historical trends.

**All of those data points from just one NVP and ECB Owner names set by configuration!**

# The Value of Using Name-Value Pairs

- Start small, with a single NVP and ECB Owner Name Hi (set by configuration) and explore the value of the data.
- Experiment with creating the views and analysis you desire for those two dimensions. Go beyond what we've described.

For example, for a specified breakdown (MsgType or OwnerHi) show different NVP values and their existence time in descending order over time.

- Add in additional horizontal NVPs (values do not change through the life of the processing), ECB Owner Mid, and vertical NVPs such as ECBPurpose or other NVPs that change through the life of processing at collection boundaries (ECB Exit and Owner Name Change).

# The Value of Using Name-Value Pairs

- Experiment with other NVPs such as horizontal NVPs that categorize message input parameters. For example the name SearchRadius could have the values: Small, Medium, Large, and Extreme.
- When you have a better feel for how you want to use NVPC, standardize your practices across your code base.
- Consider setting NVPs in your middle ware packages such that you do not have change your applications down stream.

# The Value of Using Name-Value Pairs

You may be surprised by what you learn about your applications. We were!

# Thank you!

Jesus Galvez  
z/TPF Development



# Trademarks

IBM, the IBM logo, ibm.com and Rational are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

## Notes

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.