

Modernize Your Data Sources: An IBM zPCR Story

Presenting: Kiri Nicholson (kiri.alice.nicholson@ibm.com)

Developed by: Shawn Lundvall (lundvall@us.ibm.com),

Alexander Giemsa (alexander.giemsa@ibm.com)

IBM Washington Systems Center

Modernize Your Data Sources: An IBM zPCR Story

Presenting: Kiri Nicholson (kiri.alice.nicholson@ibm.com)

Developed by: Shawn Lundvall (lundvall@us.ibm.com),

Alexander Giemsa (alexander.giemsa@ibm.com)

IBM  **Worldwide**  Systems Center

Agenda

- Review of z/OS Data Gatherer REST Services
- The zPCR Use Case
- Flow from z/OS Data Gatherer to zPCR
- Conclusions & Review

After this session, I will...

Understand how z/OS Data Gatherer REST Services could be used to modernize my IBM Z capacity sizing with IBM zPCR.

After this session, I will...

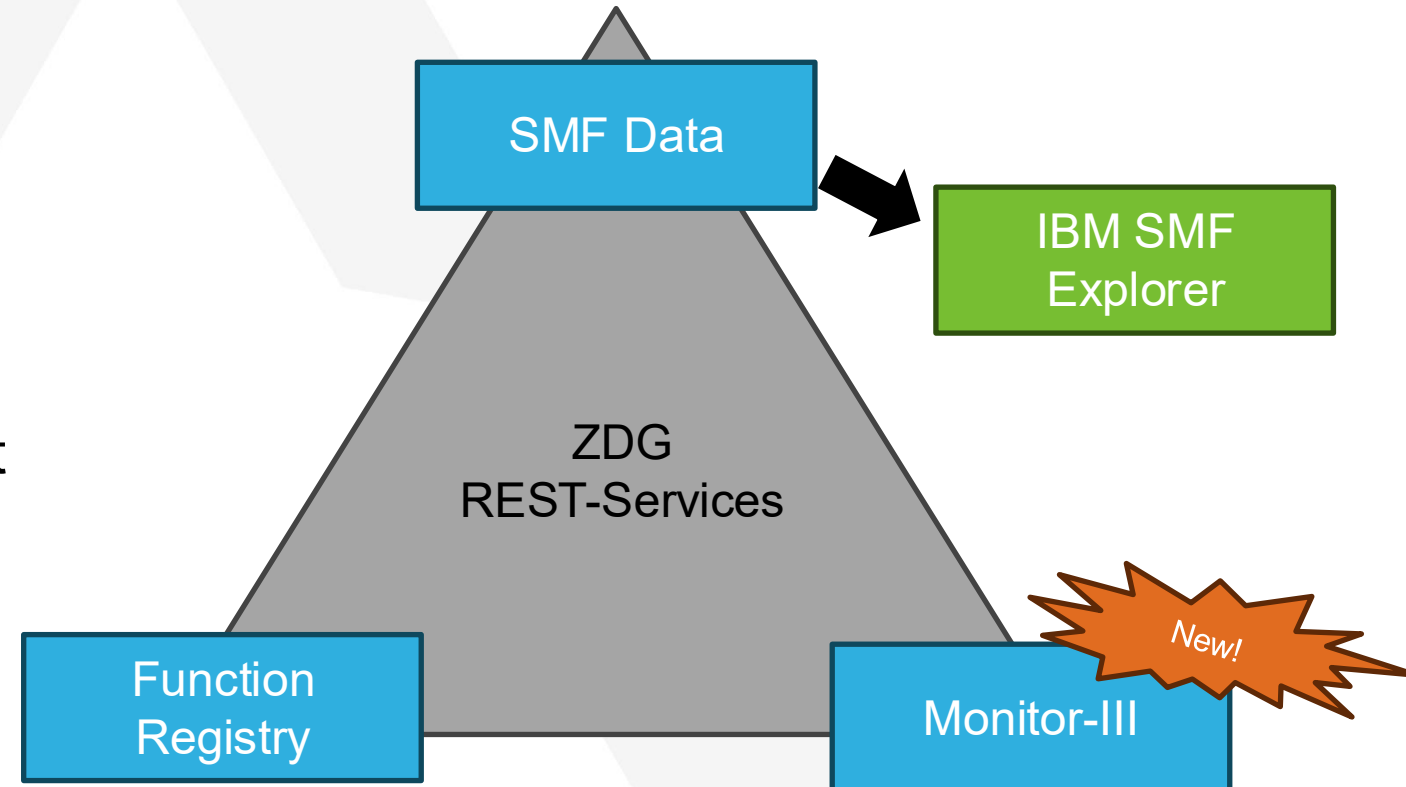
Understand how z/OS Data Gatherer REST Services could be used to modernize my IBM Z capacity sizing with IBM zPCR.



z/OS DATA GATHERER REST SERVICES

z/OS Data Gatherer REST Services

- SMF REST Services
 - Types 7x, 99.x, 113, 30
 - Backend to IBM SMF Explorer
- IBM Function Registry
 - Registered Functions on the Host
- Monitor-III REST Services
 - Short-term reporting data
- Latest Update OA68419



z/OS Data Gatherer REST Services

- SMF REST Services
 - Types 7x, 99.x, 113, 30
 - Backend to IBM SMF Explorer
- IBM Function Registry
 - Registered Functions on the Host
- Monitor-III REST Services
 - Short-term reporting data
- Latest Update OA68419

SMF Data

IBM SMF Explorer!

A word about SMF Explorer

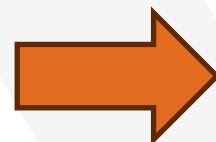
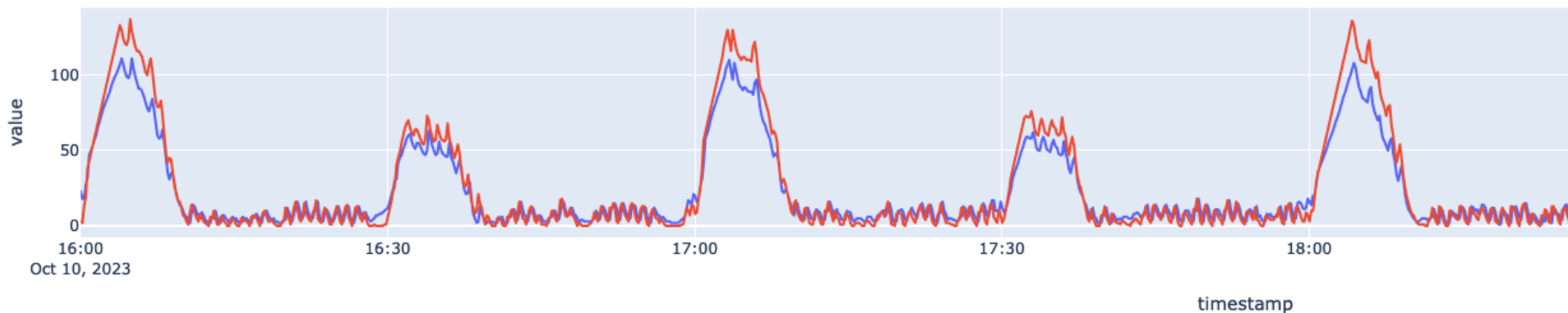
```
import smfexplorer
from smfexplorer.fields import SMF99S2
from smfexplorer import ASC, DESC

import plotly.express as px
```

```
ctx = smfexplorer.new_context('RMF.TEST.SMF992.AIDATA.SHARE')
```

```
fields = [SMF99S2.timestamp, SMF99S2.srv_class_name,\
          SMF99S2.ai_prediction, SMF99S2.qdata_server_active]
data = ctx.request(fields).where(SMF99S2.srv_class_name == 'AIBATCH1')\
      .sort(DESC(SMF99S2.timestamp)).run()
```

```
px.line(data, x='timestamp',\
        y=['ai_prediction', 'qdata_server_active'])
```

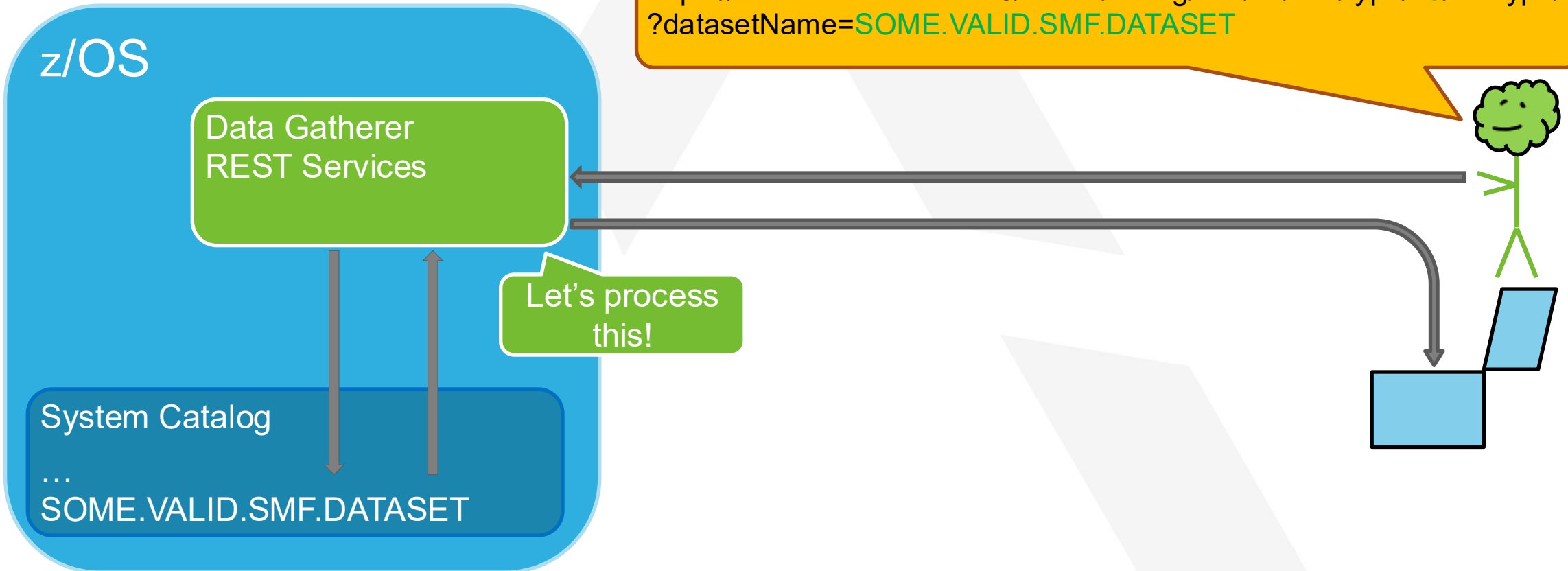


Try it out yourself for free with [IBM Z Trial](#) !

	timestamp	srv_class_name	ai_prediction	qdata_server_active
0	2023-10-10 19:47:35.770	AIBATCH1	5	0
1	2023-10-10 19:47:25.760	AIBATCH1	8	2
2	2023-10-10 19:47:15.750	AIBATCH1	12	4
3	2023-10-10 19:47:05.750	AIBATCH1	14	10
4	2023-10-10 19:46:55.750	AIBATCH1	12	15

Simplified request flow

`https://zos.is.sweet.com:445/zosmf/zosdg/smf/v1/smf/type/70/subtype/1?datasetName=SOME.VALID.SMF.DATASET`



One way

```
001 Record Number 2: SMF Record Type 70(1) - RMF CPU Activity
002 =====
003
004 -> SMF record header
005 =====
006
007 SMF record length      : 32396
008 SMF segment descriptor : '0000'X
009 SMF system indicator   : '11011111'B
010 SMF record type       : 70
011 SMF record time       : 03:59:31
012 SMF record date       : 22.054
013 SMF system id        :
014 SMF subsystem id     : RMF
015 SMF record subtype    : 1
016
017 -> RMF header extension
018 =====
019
020 Number of triplets     : 9
021
022 Section 1 offset      : '00000064'X
023 Section 1 length     : '009C'X
024 Section 1 number     : 1
025
026 Section 2 offset      : '00000100'X
027 Section 2 length     : '0158'X
```

```
00058 -> RMF Product Section (1)
00059 =====
00060
00061 #1: +0000: 797FD9D4 C6404040 40400034 431F0122 *'"RMF      C  *
00062      +0010: 054F1500 003F0000 00000384 00001001 * |          d  *
00063      +0020: 40404040 0001000F E9E5F0F2 F0F5F0F0 *           ZV020500*
00064      +0030: 03770C8E DB1B6F20 7CC00000 00000D69 * i l ! ? @ {  Z*
00065      +0040: 3A400000 00000068 00340001 03840000 *           Y    d  *
00066      +0050: DB1B6F20 7CC00000 E2E8E2C4 D7D3C5E7 * ! ? @ {      *
00067      +0060: E2E8E2C6 40404040 00020001 00000010 *SYSF          *
00068      +0070: 00040009 00000000 00010001 00010001 *              *
00069      +0080: 00050001 00010001 003C0001 01D90001 *              R  *
00070      +0090: 00060001 00000000 00060001          *              *
00071
00072 -> CPU Control Section (1)
00073 =====
00074
00075 #1: +0000: 39060069 18990000 F7F5F940 40404040 * Z r 759  *
00076      +0010: 40404040 40404040 00B00000 000001C6 *           a    F*
```

One way

```
001 Record Number 2: SMF Record Type 70(1) - RMF CPU Activity
002 =====
003
004 -> SMF record header
005 =====
006
007 SMF record length      : 32396
008 SMF segment descriptor : '0000'X
009 SMF system indicator  : '11011111'B
010 SMF record type       : 70
011 SMF record time      : 03:59:31
012 SMF record date      : 22.054
013 SMF system id        :
014 SMF subsystem id     : RMF
015 SMF record subtype   : 1
016
017 -> RMF header extension
018 =====
019
020 Number of triplets    : 9
021
022 Section 1 offset     : '00000064'X
023 Section 1 length    : '009C'X
024 Section 1 number    : 1
025
026 Section 2 offset     : '00000100'X
027 Section 2 length    : '0158'X
```

```
00058 -> RMF Product Section (1)
00059 =====
00060
00061 #1: +0000: 797FD9D4 C6404040 40400034 431F0122 *'"RMF      C  *
00062 +0010: 054F1500 003F0000 00000384 00001001 * |          d  *
00063 +0020: 40404040 0001000F E9E5F0F2 F0F5F0F0 *           ZV020500*
00064 +0030: 03770C8E DB1B6F20 7CC00000 00000D69 * i l ! ? @ {  Z*
00065 +0040: 3A400000 00000068 00340001 03840000 *           Y    d  *
00066 +0050: DB1B6F20 7CC00000 E2E8E2C4 D7D3C5E7 *! ? @ {      *
00067 +0060: E2E8E2C6 40404040 00020001 00000010 *SYSF          *
00068 +0070: 00040009 00000000 00010001 00010001 *           *
00069 +0080: 00050001 00010001 003C0001 01D90001 *           R  *
00070 +0090: 00060001 00000000 00060001          *           *
00071
00072 -> CPU Control Section (1)
00073 =====
00074
00075 #1: +0000: 39060069 18990000 F7F5F940 40404040 * Z r 759  *
00076 +0010: 40404040 40404040 00B00000 000001C6 *           a  F*
```

One way

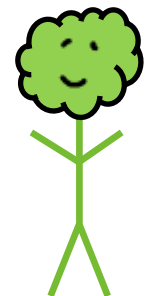
```
001 Record Number 2: SMF Record Type 70(1) - RMF CPU Activity
002 =====
003
004 -> SMF record header
005 =====
006
007 SMF record length      : 32396
008 SMF segment descriptor : '0000'X
009 SMF system indicator  : '11011111'B
010 SMF record type       : 70
011 SMF record time      : 03:59:31
012 SMF record date      : 22.054
013 SMF system id        :
014 SMF subsystem id     : RMF
015 SMF record subtype   : 1
016
017 -> RMF header extension
018 =====
019
020 Number of triplets    : 9
021
022 Section 1 offset      : '00000064'X
023 Section 1 length     : '009C'X
024 Section 1 number     : 1
025
026 Section 2 offset      : '00000100'X
027 Section 2 length     : '0158'X
```

```
00058 -> RMF Product Section (1)
00059 =====
00060
00061 #1: +0000: 797FD9D4 C6404040 40400034 431F0122 *'"RMF      C  *
00062 +0010: 054F1500 003F0000 00000384 00001001 * |          d  *
00063 +0020: 40404040 0001000F E9E5F0F2 F0F5F0F0 *           ZV020500*
00064 +0030: 03770C8E DB1B6F20 7CC00000 00000D69 * i l! ? @{    Z*
00065 +0040: 3A400000 00000068 00340001 03840000 *           Y    d  *
00066 +0050: DB1B6F20 7CC00000 E2E8E2C4 D7D3C5E7 *! ? @{      *
00067 +0060: E2E8E2C6 40404040 00020001 00000010 *SYSF      *
00068 +0070: 00040009 00000000 00010001 00010001 *           *
00069 +0080: 00050001 00010001 003C0001 01D90001 *           R  *
00070 +0090: 00060001 00000000 00060001          *           *
00071
00072 -> CPU Control Section (1)
00073 =====
00074
00075 #1: +0000: 39060069 18990000 F7F5F940 40404040 * Z r 759  *
00076 +0010: 40404040 40404040 00B00000 000001C6 *           r  F*
```

Another way!

```
▶ 0:      {...}
▶ 1:      {...}
▼ 2:
SMF70LEN:      1520
SMF70SEG:      0
SMF70FLG:      "11011110"
SMF70RRF:      true
SMF70SUT:      true
SMF70V4:      true
SMF70ESA:      true
SMF70VXA:      true
SMF70OS:       true
SMF70BFY:      false
SMF70RTY:      70
SMF70TME:      "14:59:00.01"
SMF70DTE:      "2017-06-21"
SMF70SID:
SMF70SSI:      "RMF "
SMF70STY:      1
SMF70TRN:      9
SMF70PRS:      100
SMF70PRL:      104
SMF70PRN:      1
SMF70CCS:      204
SMF70CCL:      344
SMF70CCN:      1
SMF70CPS:      548
SMF70CPL:      92
SMF70CPN:      2
SMF70ASS:      732
SMF70ASL:      788
SMF70ASN:      1
SMF70BCS:      1520
SMF70BCL:      84
SMF70BCN:      0
SMF70BVS:      1520
```

```
SMF70TNS:      0
SMF70TNL:      0
SMF70TNN:      0
▶ smf70Subtype1ProductSection:  {...}
▶ smf70Subtype1AsidArea:      {...}
▼ smf70Subtype1CpuControl:
  SMF70MOD:      "2964"
  SMF70VER:      255
  SMF70BNP:      0
  SMF70INB:      "00000000"
  SMF70DIF:      false
  SMF70NPC:      false
  SMF70TSC:      false
  SMF70PHY:      false
  SMF70DGE:      false
  SMF70VMG:      false
  SMF70STF:      "10011000"
  SMF70STS:      true
  SMF70ADC:      false
  SMF70WUC:      false
  SMF70RCU:      true
  SMF70HWV:      true
  SMF70PTC:      false
  SMF70PLC:      false
  SMF70GAV:      false
  SMF70GTS:      0
  SMF70MDL:      "757"
  SMF70DSA:      12
  SMF70IFA:      0
  SMF70CPA:      506
  SMF70WLA:      1024
  SMF70LAC:      2
  SMF70HWM:      "N63"
  SMF70SUP:      0
  SMF70GJT:      "02:00:00"
  SMF70POM:      "02"
```



Result: JSON

```
▶ 0: {}
▶ 1: {}
▼ 2:
  SMF70LEN: 1520
  SMF70SEG: 0
  SMF70FLG: "11011110"
  SMF70RRF: true
  SMF70SUT: true
  SMF70V4: true
  SMF70ESA: true
  SMF70VXA: true
  SMF700S: true
  SMF70BFY: false
  SMF70RTY: 70
  SMF70TME: "14:59:00.01"
  SMF70DTE: "2017-06-21"
  SMF70SID:
  SMF70SSI: "RMF "
  SMF70STY: 1
  SMF70TRN: 9
  SMF70PRS: 100
  SMF70PRL: 104
  SMF70PRN: 1
  SMF70CCS: 204
  SMF70CCL: 344
  SMF70CCN: 1
  SMF70CPS: 548
  SMF70CPL: 92
  SMF70CPN: 2
  SMF70ASS: 732
  SMF70ASL: 788
  SMF70ASN: 1
  SMF70BCS: 1520
  SMF70BCL: 84
  SMF70BCN: 0
  SMF70BVS: 1520
```

```
SMF70TNS: 0
SMF70TNL: 0
SMF70TNN: 0
▶ smf70Subtype1ProductSection: {}
▶ smf70Subtype1AsidArea: {}
▼ smf70Subtype1CpuControl:
  SMF70MOD: "2964"
  SMF70VER: 255
  SMF70BNP: 0
  SMF70INB: "00000000"
  SMF70DIF: false
  SMF70NPC: false
  SMF70TSC: false
  SMF70PHY: false
  SMF70DGE: false
  SMF70VMG: false
  SMF70STF: "10011000"
  SMF70STS: true
  SMF70ADC: false
  SMF70WUC: false
  SMF70RCU: true
  SMF70HWV: true
  SMF70PTC: false
  SMF70PLC: false
  SMF70GAV: false
  SMF70GTS: 0
  SMF70MDL: "757"
  SMF70DSA: 12
  SMF70IFA: 0
  SMF70CPA: 506
  SMF70WLA: 1024
  SMF70LAC: 2
  SMF70HWM: "N63"
  SMF70SUP: 0
  SMF70GJT: "02:00:00"
  SMF70POM: "02"
```

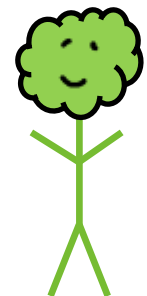


Result: JSON

Indexing

```
0: {}
1: {}
2:
  SMF70LEN: 1520
  SMF70SEG: 0
  SMF70FLG: "11011110"
  SMF70RRF: true
  SMF70SUT: true
  SMF70V4: true
  SMF70ESA: true
  SMF70VXA: true
  SMF700S: true
  SMF70BFY: false
  SMF70RTY: 70
  SMF70TME: "14:59:00.01"
  SMF70DTE: "2017-06-21"
  SMF70SID:
  SMF70SSI: "RMF "
  SMF70STY: 1
  SMF70TRN: 9
  SMF70PRS: 100
  SMF70PRL: 104
  SMF70PRN: 1
  SMF70CCS: 204
  SMF70CCL: 344
  SMF70CCN: 1
  SMF70CPS: 548
  SMF70CPL: 92
  SMF70CPN: 2
  SMF70ASS: 732
  SMF70ASL: 788
  SMF70ASN: 1
  SMF70BCS: 1520
  SMF70BCL: 84
  SMF70BCN: 0
  SMF70BVS: 1520
```

```
SMF70TNS: 0
SMF70TNL: 0
SMF70TNN: 0
  ▶ smf70Subtype1ProductSection: {}
  ▶ smf70Subtype1AsidArea: {}
  ▼ smf70Subtype1CpuControl:
    SMF70MOD: "2964"
    SMF70VER: 255
    SMF70BNP: 0
    SMF70INB: "00000000"
    SMF70DIF: false
    SMF70NPC: false
    SMF70TSC: false
    SMF70PHY: false
    SMF70DGE: false
    SMF70VMG: false
    SMF70STF: "10011000"
    SMF70STS: true
    SMF70ADC: false
    SMF70WUC: false
    SMF70RCU: true
    SMF70HWV: true
    SMF70PTC: false
    SMF70PLC: false
    SMF70GAV: false
    SMF70GTS: 0
    SMF70MDL: "757"
    SMF70DSA: 12
    SMF70IFA: 0
    SMF70CPA: 506
    SMF70WLA: 1024
    SMF70LAC: 2
    SMF70HWM: "N63"
    SMF70SUP: 0
    SMF70GJT: "02:00:00"
    SMF70POM: "02"
```



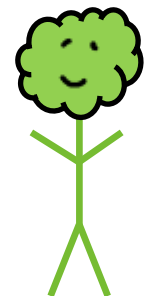
Result: JSON

Indexing

0:	{...}
1:	{...}
2:	{...}
SMF70LEN:	1520
SMF70SEGS:	0
SMF70FLG:	"11011110"
SMF70RRF:	true
SMF70SUT:	true
SMF70V4:	true
SMF70ESA:	true
SMF70VXA:	true
SMF700S:	true
SMF70BFY:	false
SMF70RTY:	70
SMF70TME:	"14:59:00.01"
SMF70DTE:	"2017-06-21"
SMF70SID:	
SMF70SSI:	"RMF "
SMF70STY:	1
SMF70TRN:	9
SMF70PRS:	100
SMF70PRL:	104
SMF70PRN:	1
SMF70CCS:	204
SMF70CCL:	344
SMF70CCN:	1
SMF70CPS:	548
SMF70CPL:	92
SMF70CPN:	2
SMF70ASS:	732
SMF70ASL:	788
SMF70ASN:	1
SMF70BCS:	1520
SMF70BCL:	84
SMF70BCN:	0
SMF70BVS:	1520

SMF70TNS:	0
SMF70TNL:	0
SMF70TNN:	0
smf70Subtype1ProductSection:	{...}
smf70Subtype1AsidArea:	{...}
smf70Subtype1CpuControl:	
SMF70MOD:	"2964"
SMF70VER:	255
SMF70BNP:	0
SMF70INB:	"00000000"
SMF70DIF:	false
SMF70NPC:	false
SMF70TSC:	false
SMF70PHY:	false
SMF70DGE:	false
SMF70VMG:	false
SMF70STF:	"10011000"
SMF70STS:	true
SMF70ADC:	false
SMF70WUC:	false
SMF70RCU:	true
SMF70HWV:	true
SMF70PTC:	false
SMF70PLC:	false
SMF70GAV:	false
SMF70GTS:	0
SMF70MDL:	"757 "
SMF70DSA:	12
SMF70IFA:	0
SMF70CPA:	506
SMF70WLA:	1024
SMF70LAC:	2
SMF70HWM:	"N63 "
SMF70SUP:	0
SMF70GJT:	"02:00:00"
SMF70POM:	"02 "

Formatting



Result: JSON

Indexing

0:	{...}
1:	{...}
2:	
SMF70LEN:	1520
SMF70SEG:	0
SMF70FLG:	"11011110"
SMF70RRF:	true
SMF70SUT:	true
SMF70V4:	true
SMF70ESA:	true
SMF70VXA:	true
SMF700S:	true
SMF70BFY:	false
SMF70RTY:	70
SMF70TME:	"14:59:00.01"
SMF70DTE:	"2017-06-21"
SMF70SID:	
SMF70SSI:	"RMF "
SMF70STY:	1
SMF70TRN:	9
SMF70PRS:	100
SMF70PRL:	104
SMF70PRN:	1
SMF70CCS:	204
SMF70CCL:	344
SMF70CCN:	1
SMF70CPS:	548
SMF70CPL:	92
SMF70CPN:	2
SMF70ASS:	732
SMF70ASL:	788
SMF70ASN:	1
SMF70BCS:	1520
SMF70BCL:	84
SMF70BCN:	0
SMF70BVS:	1520

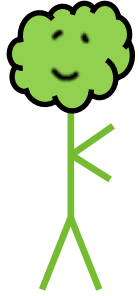
SMF70TNS:	0
SMF70TNL:	0
SMF70TNN:	0
smf70Subtype1ProductSection:	{...}
smf70Subtype1AsidArea:	{...}
smf70Subtype1CpuControl:	
SMF70MOD:	"2964"
SMF70VER:	255
SMF70BNP:	0
SMF70INB:	"00000000"
SMF70DIF:	false
SMF70NPC:	false
SMF70TSC:	false
SMF70PHY:	false
SMF70DGE:	false
SMF70VMG:	false
SMF70STF:	"10011000"
SMF70STS:	true
SMF70ADC:	false
SMF70WUC:	false
SMF70RCU:	true
SMF70HWV:	true
SMF70PTC:	false
SMF70PLC:	false
SMF70GAV:	false
SMF70GTS:	0
SMF70MDL:	"757"
SMF70DSA:	12
SMF70IFA:	0
SMF70CPA:	506
SMF70WLA:	1024
SMF70LAC:	2
SMF70HWM:	"N63"
SMF70SUP:	0
SMF70GJT:	"02:00:00"
SMF70POM:	"02"

Formatting

Hierarchy and 1-to-1 relationships



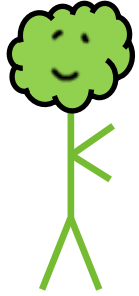
Result: JSON



```
SMF70COS: 1520
SMF70COL: 16
SMF70CON: 0
SMF70TNS: 0
SMF70TNL: 0
SMF70TNN: 0
▶ smf70Subtype1ProductSection: {...}
▶ smf70Subtype1AsidArea: {...}
▶ smf70Subtype1CpuControl: {...}
▼ smf70Subtype1CpuData:
  ▼ 0:
    SMF70WAT: "00:00:57.093"
    SMF70CID: 0
    SMF70CNF: "00000001"
    SMF70MTI: false
    SMF70DCI: false
    SMF70PAR: false
    SMF70VAC: false
    SMF70STA: true
    SMF70SER: "0A5327"
    SMF70TYP: 0
    SMF70SLH: 3506
    SMF70TPI: 0
    SMF70VFS: 0
    SMF70V: "00000000"
    SMF70VON: false
    SMF70PAT: "00:00:00"
    SMF70TCB: 9608
    SMF70SRB: 9937
    SMF70NIO: 3554
    SMF70SIG: 2111
    SMF70WTD: 14629
    SMF70WTS: 0
    SMF70WTU: 0
    SMF70WTI: 0
  ▶ 1: {...}
```

```
R782EMR: 1584291840
▼ smf78Subtype2R782sqau:
  VSDBMIN: 258048
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 258048
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 1548288
  VSDAMIN: 20787200
  VSDANTME: "2017-06-21T14:59:09.734"
  VSDAMAX: 20795392
  VSDAXTME: "2017-06-21T14:58:59.249"
  VSDATOTL: 124731392
▼ smf78Subtype2R782csau:
  VSDBMIN: 274432
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 274432
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 1646592
  VSDAMIN: 28516352
  VSDANTME: "2017-06-21T14:59:09.734"
  VSDAMAX: 28622848
  VSDAXTME: "2017-06-21T14:58:59.249"
  VSDATOTL: 171204608
▶ smf78Subtype2R782csaf: {...}
▼ smf78Subtype2R782cslf:
  VSDBMIN: 3473408
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 3473408
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 20840448
  VSDAMIN: 286642176
  VSDANTME: "2017-06-21T14:58:59.249"
  VSDAMAX: 286715904
  VSDAXTME: "2017-06-21T14:59:09.734"
  VSDATOTL: 1720221696
▶ smf78Subtype2R782csal: {...}
```

Result: JSON

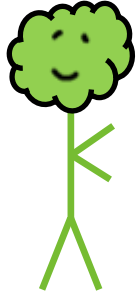


```
SMF70C0S: 1520
SMF70COL: 16
SMF70CON: 0
SMF70TNS: 0
SMF70TNL: 0
SMF70TNN: 0
▶ smf70Subtype1ProductSection: {...}
▶ smf70Subtype1AsidArea: {...}
▶ smf70Subtype1CpuControl: {...}
▼ smf70Subtype1CpuData:
  ▼ 0:
    SMF70WAT: "00:00:57.093"
    SMF70CID: 0
    SMF70CNF: "00000001"
    SMF70MTI: false
    SMF70DCI: false
    SMF70PAR: false
    SMF70VAC: false
    SMF70STA: true
    SMF70SER: "0A5327"
    SMF70TYP: 0
    SMF70SLH: 3506
    SMF70TPI: 0
    SMF70VFS: 0
    SMF70V: "00000000"
    SMF70VON: false
    SMF70PAT: "00:00:00"
    SMF70TCB: 9608
    SMF70SRB: 9937
    SMF70NIO: 3554
    SMF70SIG: 2111
    SMF70WTD: 14629
    SMF70WTS: 0
    SMF70WTU: 0
    SMF70WTI: 0
  ▶ 1: {...}
```

```
R782EMR: 1584291840
▼ smf78Subtype2R782sqau:
  VSDBMIN: 258048
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 258048
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 1548288
  VSDAMIN: 20787200
  VSDANTME: "2017-06-21T14:59:09.734"
  VSDAMAX: 20795392
  VSDAXTME: "2017-06-21T14:58:59.249"
  VSDATOTL: 124731392
▼ smf78Subtype2R782csau:
  VSDBMIN: 274432
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 274432
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 1646592
  VSDAMIN: 28516352
  VSDANTME: "2017-06-21T14:59:09.734"
  VSDAMAX: 28622848
  VSDAXTME: "2017-06-21T14:58:59.249"
  VSDATOTL: 171204608
▶ smf78Subtype2R782csaf: {...}
▼ smf78Subtype2R782cslf:
  VSDBMIN: 3473408
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 3473408
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 20840448
  VSDAMIN: 286642176
  VSDANTME: "2017-06-21T14:58:59.249"
  VSDAMAX: 286715904
  VSDAXTME: "2017-06-21T14:59:09.734"
  VSDATOTL: 1720221696
▶ smf78Subtype2R782csal: {...}
```

1-to-many
relationships

Result: JSON



```
SMF70COS: 1520
SMF70COL: 16
SMF70CON: 0
SMF70TNS: 0
SMF70TNL: 0
SMF70TNN: 0
▶ smf70Subtype1ProductSection: {...}
▶ smf70Subtype1AsidArea: {...}
▶ smf70Subtype1CpuControl: {...}
▼ smf70Subtype1CpuData:
  ▼ 0:
    SMF70WAT: "00:00:57.093"
    SMF70CID: 0
    SMF70CNF: "00000001"
    SMF70MTI: false
    SMF70DCI: false
    SMF70PAR: false
    SMF70VAC: false
    SMF70STA: true
    SMF70SER: "0A5327"
    SMF70TYP: 0
    SMF70SLH: 3506
    SMF70TPI: 0
    SMF70VFS: 0
    SMF70V: "00000000"
    SMF70VON: false
    SMF70PAT: "00:00:00"
    SMF70TCB: 9608
    SMF70SRB: 9937
    SMF70NIO: 3554
    SMF70SIG: 2111
    SMF70WTD: 14629
    SMF70WTS: 0
    SMF70WTU: 0
    SMF70WTI: 0
  ▶ 1: {...}
```

1-to-many relationships

```
R782EMR: 1584291840
▼ smf78Subtype2R782sqau:
  VSDBMIN: 258048
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 258048
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 1548288
  VSDAMIN: 20787200
  VSDANTME: "2017-06-21T14:59:09.734"
  VSDAMAX: 20795392
  VSDAXTME: "2017-06-21T14:58:59.249"
  VSDATOTL: 124731392
▼ smf78Subtype2R782csau:
  VSDBMIN: 274432
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 274432
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 1646592
  VSDAMIN: 28516352
  VSDANTME: "2017-06-21T14:59:09.734"
  VSDAMAX: 28622848
  VSDAXTME: "2017-06-21T14:58:59.249"
  VSDATOTL: 171204608
▶ smf78Subtype2R782csaf: {...}
▼ smf78Subtype2R782cslf:
  VSDBMIN: 3473408
  VSDBNTME: "2017-06-21T14:58:59.249"
  VSDBMAX: 3473408
  VSDBXTME: "2017-06-21T14:58:59.249"
  VSDBTOTL: 20840448
  VSDAMIN: 286642176
  VSDANTME: "2017-06-21T14:58:59.249"
  VSDAMAX: 286715904
  VSDAXTME: "2017-06-21T14:59:09.734"
  VSDATOTL: 1720221696
▶ smf78Subtype2R782csal: {...}
```

78-2 implicit structures

Specific endpoint per (sub-)type

SMF 70 RMF Processor Activity ^

GET `/v1/smf/type/70/subtype/1` CPU, PR/SM, and ICF Activity ∨

GET `/v1/smf/type/70/subtype/2` Cryptographic Hardware Activity ∨

SMF 71 ^

GET `/v1/smf/type/71/subtype/1` RMF Paging Activity ∨

SMF 72 Workload Activity, Storage Data, and Serialization Delay ^

GET `/v1/smf/type/72/subtype/3` Workload Activity ∨

GET `/v1/smf/type/72/subtype/4` Storage Data ∨

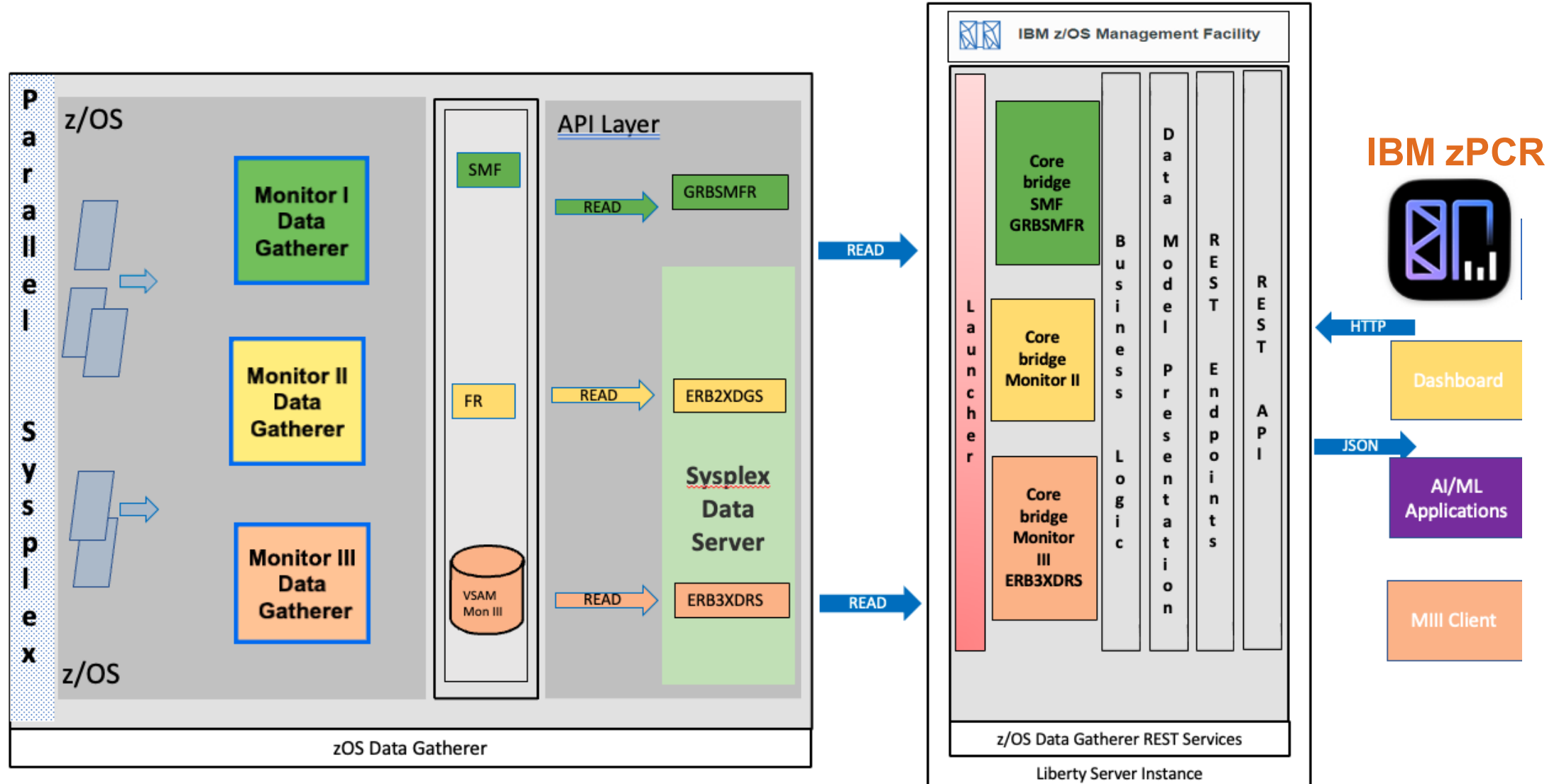
GET `/v1/smf/type/72/subtype/5` Serialization Delay ∨

... up to SMF 79 plus SMF 30

SMF 99 subtypes 1, 2, 6, 12 and 14

SMF 113 subtypes 1 and 2

REST Services deployment



Data Flow

```

Command ==>
000012 1Rec-Num Type RecLn SMFDate SMFTime RMFDate RMFTime
000013 -----
000014 1 002 18 2023.292 12:38:18
000015 2 002 18 2021.040 02:35:02
000016 3 070.001 27032 2020.062 00:00:00 2020.061 23:30:00
000017 4 072.003

```

```

Command ==>
000018 5 072.003 000257 -> PR/SM Partition Data Section (15)
000019 6 072.003 000258 =====
000020 7 072.003 000259
000021 8 072.003 000260 #1: +0000: C1D8C6E3 40404040 01040000 00000000 *AQFT *
000022 9 072.003 000261 +0010: 00000000 D4C3D3E7 C3C6F0F1 C1D8C6E3 * MCLXCF01AQFT*
000023 10 072.003 000262 +0020: 40404040 00000000 0000F000 00000000 * 0 *
000024 11 072.003 000263 +0030: 00000000 00000000 A0000301 00000000 * μ *
000025 12 072.003 000264 +0040: 00000000 00000000 00000000 00000000 * *
000265 +0050: 00000000 * *
000266
000267 #2: +0000: E5C9C3E3 C5E2E340 02040005 00000000 *VICTEST *
000268 +0010: 00000000 00000000 00000000 E5C9C3E3 * VICT*
000269 +0020: E2E34040 00000000 00002800 00000000 *ST *
000270 +0030: 00000000 00000000 A0000200 00000000 * μ *
000271 +0040: 00000000 00000000 00000000 00000000 * *
000272 +0050: 00000000 * *
000273
000274 #3: +0000: E5D4E3D6 D6D3F140 0304001E 00000012 *VMTOOL1 *
000275 +0010: 00000000 00000000 00000000 E5D4E3D6 * VMTO*

```

Data Gatherer REST Services



```

Command ==>
000012 1Rec-Num Type RecLn SMFDate SMFTime RMFDate RMFTime
000013 -----
000014 1 002 18 2023.292 12:38:18
000015 2 002 18 2021.040 02:35:02
000016 3 070.001 27032 2020.062 00:00:00 2020.061 23:30:00
000017 4 072.003

```

```

Command ==>
000257 -> PR/SM Partition Data Section (15)
000258 =====
000259
000260 #1: +0000: C1D8C6E3 40404040 01040000 00000000 *AQFT *
000261 +0010: 00000000 D4C3D3E7 C3C6F0F1 C1D8C6E3 * MCLXCF01AQFT*
000262 +0020: 40404040 00000000 0000F000 00000000 * 0 *
000263 +0030: 00000000 00000000 A0000301 00000000 * μ *
000264 +0040: 00000000 00000000 00000000 00000000 * *
000265 +0050: 00000000 * *
000266
000267 #2: +0000: E5C9C3E3 C5E2E340 02040005 00000000 *VICTEST *
000268 +0010: 00000000 00000000 00000000 E5C9C3E3 * VICT*
000269 +0020: E2E34040 00000000 00002800 00000000 *ST *
000270 +0030: 00000000 00000000 A0000200 00000000 * μ *
000271 +0040: 00000000 00000000 00000000 00000000 * *
000272 +0050: 00000000 * *
000273
000274 #3: +0000: E5D4E3D6 D6D3F140 0304001E 00000012 *VMTOOL1 *
000275 +0010: 00000000 00000000 00000000 E5D4E3D6 * VMTO*

```

```

▼ 3:
SMF70TME: "00:00:00.03"
SMF70DTE: "2020-03-02"
SMF70SID: "AQFT"
▼ smf70Subtype1PrSmPartitionData:
▼ 0:
SMF70LPM: "AQFT "
SMF70LPN: 1
SMF70BDN: 13
▼ 1:
SMF70LPM: "VICTEST "
SMF70LPN: 2
SMF70BDN: 5
▼ 2:
SMF70LPM: "VMTOOL1 "
SMF70LPN: 3
SMF70BDN: 30
▼ 3:
SMF70LPM: "AQAWARE1"
SMF70LPN: 4
SMF70BDN: 1
▼ 4:
SMF70LPM: "AQCF1 "
SMF70LPN: 5
SMF70BDN: 1
▼ 5:
SMF70LPM: "AQHO "
SMF70LPN: 6
SMF70BDN: 7

```

Data Gatherer REST Services

```

Command ==>
000012 1Rec-Num Type RecLn SMFDate SMFTime RMFDate RMFTime
000013 -----
000014 1 002 18 2023.292 12:38:18
000015 2 002 18 2021.040 02:35:02
000016 3 070.001 27032 2020.062 00:00:00 2020.061 23:30:00
000017 4 072.003
000018 5 072.003
000019 6 072.003
000020 7 072.003
000021 8 072.003
000022 9 072.003
000023 10 072.003
000024 11 072.003
000025 12 072.003
    
```

```

Command ==>
000257 -> PR/SM Partition Data Section (15)
000258 =====
000259
000260 #1: +0000: C1D8C6E3 40404040 0104000D 00000000 *AQFT *
000261 +0010: 00000000 D4C3D3E7 C3C6F0F1 C1D8C6E3 * MCLXCF01AQFT*
000262 +0020: 40404040 00000000 0000F000 00000000 * 0 *
000263 +0030: 00000000 00000000 A0000301 00000000 * μ *
000264 +0040: 00000000 00000000 00000000 00000000 * *
000265 +0050: 00000000 * *
000266
000267 #2: +0000: E5C9C3E3 C5E2E340 02040005 00000000 *VICTEST *
000268 +0010: 00000000 00000000 00000000 E5C9C3E3 * VICT*
000269 +0020: E2E34040 00000000 00002800 00000000 *ST *
000270 +0030: 00000000 00000000 A0000200 00000000 * μ *
000271 +0040: 00000000 00000000 00000000 00000000 * *
000272 +0050: 00000000 * *
000273
000274 #3: +0000: E5D4E3D6 D6D3F140 0304001E 00000012 *VMTOOL1 *
000275 +0010: 00000000 00000000 00000000 E5D4E3D6 * VMTO
    
```



```

▼ 3:
SMF70TME: "00:00:00.03"
SMF70DTE: "2020-03-02"
SMF70SID: "AQFT"
▼ smf70Subtype1PrSmPartitionData:
▼ 0:
SMF70LPM: "AQFT "
SMF70LPN: 1
SMF70BDN: 13
▼ 1:
SMF70LPM: "VICTEST "
SMF70LPN: 2
SMF70BDN: 5
▼ 2:
SMF70LPM: "VMTOOL1 "
SMF70LPN: 3
SMF70BDN: 30
▼ 3:
SMF70LPM: "AQAWARE1"
SMF70LPN: 4
SMF70BDN: 1
▼ 4:
SMF70LPM: "AQCF1 "
SMF70LPN: 5
SMF70BDN: 1
▼ 5:
SMF70LPM: "AQHO "
SMF70LPN: 6
SMF70BDN: 7
    
```

IBM SMF Explorer

IBM zPCR



	timestamp	sid	lpar_name	system_name	sysplex_name	lpar_system_name	lpar_number	lpar_cpu_count
0	2020-03-02 00:00:00.030	AQFT	AQFT	AQFT	MCLXCF01	AQFT-AQFT	1	13
1	2020-03-02 00:00:00.030	AQFT	VICTEST	VICTST	<NA>	VICTEST-VICTST	2	5
2	2020-03-02 00:00:00.030	AQFT	VMTOOL1	VMTOOL1	<NA>	VMTOOL1-VMTOOL1	3	30
3	2020-03-02 00:00:00.030	AQFT	AQAWARE1	ZAWARE	<NA>	AQAWARE1-ZAWARE	4	1
4	2020-03-02 00:00:00.030	AQFT	AQCF1		<NA>	AQCF1-	5	1
5	2020-03-02 00:00:00.030	AQFT	AQHO	AQHO	HOXCF01	AQHO-AQHO	6	7



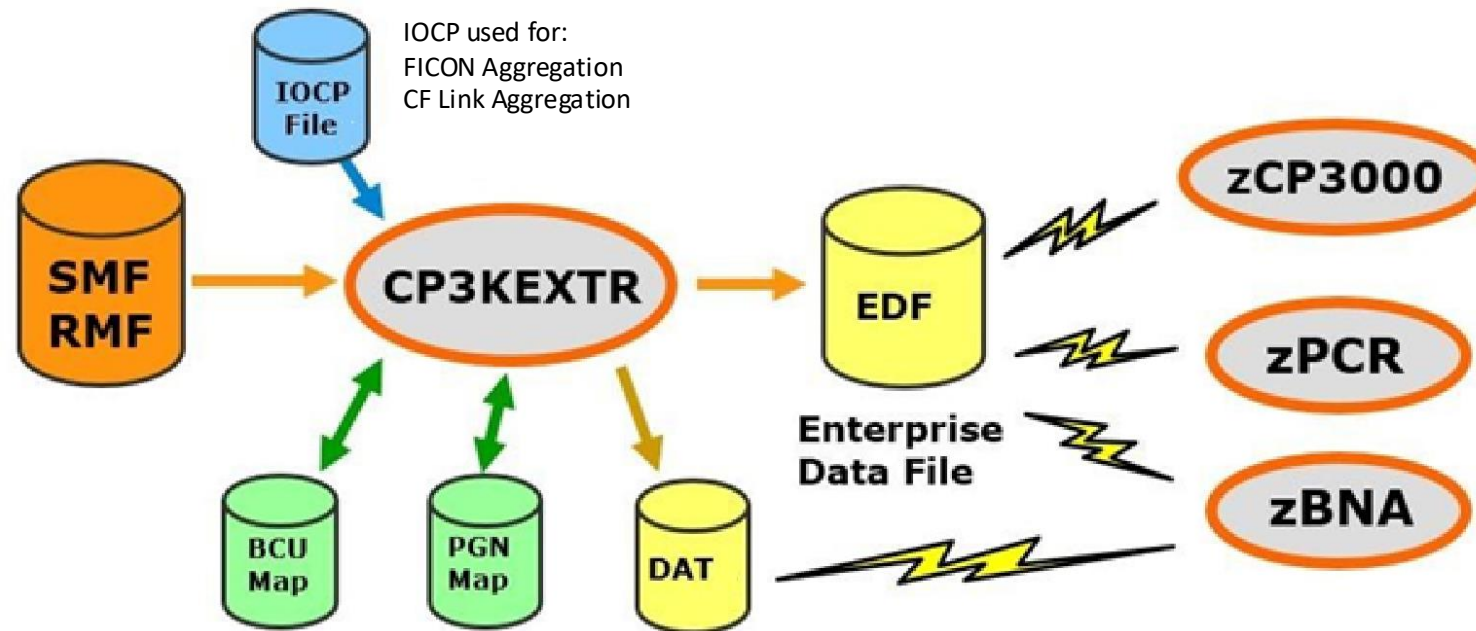
THE zPCR USE CASE

EDF File Creation for IBM zPCR

[CP3KEXTR](https://www.ibm.com/support/pages/node/6354221) Tool download website:

<https://www.ibm.com/support/pages/node/6354221>

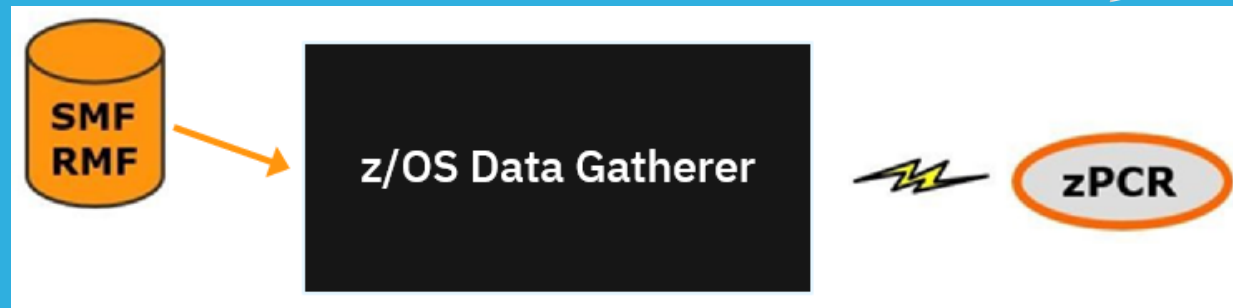
CP3KEXTR Quick Start Guide (Available on CP3KEXTR website)



* EDF (Enterprise Data File) are created using CP3KEXTR for z/OS and CP3KVMXT for z/VM

Copyright© by SHARE Association Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license. <http://creativecommons.org/licenses/by-nc-nd/3.0/>

IBM z/OS Data Gatherer for IBM zPCR PoC



Internal Early Proof of Concept

Mechanics

Control Panel [untitled]
File Edit CPcalculator Registration Documentation Help
IBM zPCR 9.7.4.json

Capacity Planning Control Panel

Study ID:

Double click on a tree branch below to access the relevant windows

- Reference-CPU
 - REF 2094-701 @ 593.00 MIPS (SI); 559.792 MIPS (MI)
- LSPR Multi-Image Processor Table
 - LSPR IBM Z General Purpose CPs
 - LSPR IBM Z IFL CPs
 - LSPR IBM LinuxONE CPs
- LPAR Configurations
 - #1 Configuration #1
No host defined

IBM z17

Manage Compare Copy & Move Partitions QuickStart Guide

LPAR Host and Partition Configuration

IBM zPCR 9.7.4.json

LPAR Configuration Capacity Planning

Based on LSPR Data for IBM Z Processors
Study ID: Not specified
#1 Configuration #1

Description:

LPAR Host Processor			Logical Partition Configuration				
Processor	Brand		CP Pool	Partition Mode	Number of		LCP:RCP Ratio
Processor	Family				Real CPs	Logical Partitions	
Processor	Model						
Speed	Class						
Maximum	CPs						
	Configured						
	RCP Pool Contention						
CP Type	Assigned	Unused					
GP							
zAAP							
zIIP							
IFL							
ICF							
Total							

Define LPAR Host Processor: Specify Host

Create Host and Partitions From: EDF JSON RMF IBM zPCR Study

Define Partitions: GP IFL ICF

Copy Partitions From: EDF RMF IBM zPCR Study

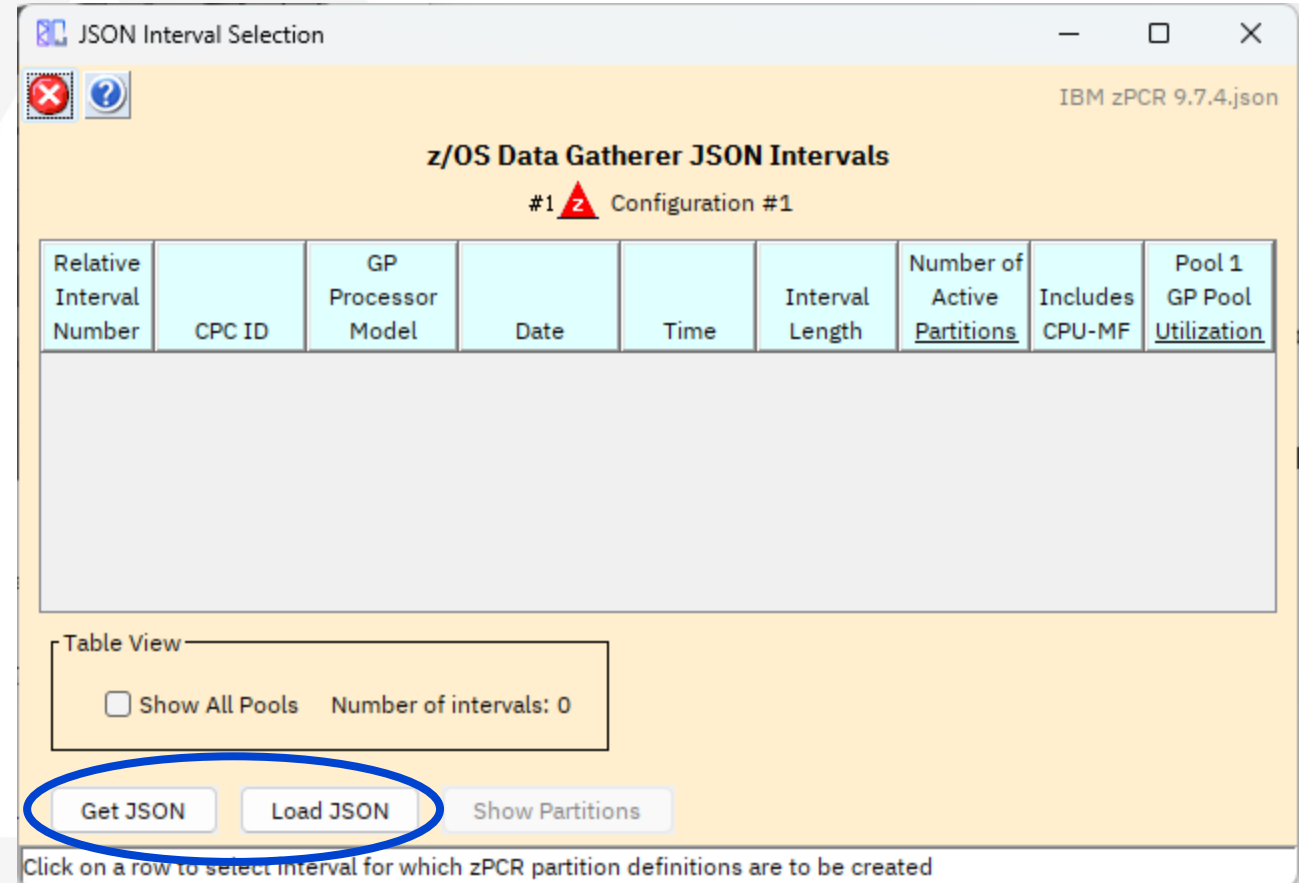
Capacity Reports: Host Summary Partition Detail Partition Utilized Capacity

Double click

New JSON button

JSON Interval Selection Window

- Mirrors existing EDF Interval Selection Window
- Use the “Get JSON” button to initiate a fetch data request from z/OS Data Gatherer
 - Creates a zip file
 - Fetch once, reuse many
- Use the “Load JSON” button to load from a previously created zip file



Setup Connection Details

z/OS Data Gatherer Connection Details

Host: <your_host_name> Port: <port_number>

Data Set Name: <data_set_name>

User: <your_user_name> Password:

Query Host Cancel

Dataset Found Providing More Parameters

z/OS Data Gatherer Connection Details

<your host name>:<port number>/<data set name>

SYS ID Interval Duration

WSC1 60 minutes

Start End

2023-11-07 00:00 2023-11-07 23:00

Get JSON Cancel

Providing More Parameters SYSID

- Click the drop-down for SYS ID
- All available SYS IDs in the SMF are presented
- Select the one for which you'd like data

z/OS Data Gatherer Connection Details

<your_host_name>:<port_number>/<data_set_name>

SYS ID Interval Duration

WSC1 60 minutes

WSC1

WSC2

WSC3

WSC4

End

00:00 2023-11-07 23:00

Get JSON Cancel

Providing More Parameters Interval Duration

- Click the drop-down for Interval Duration
- All available durations in the SMF are presented
 - Current valid options are 60, 30, 15, and 5
 - If the RMF interval is set to something greater than one of the options, it won't be shown (as seen here, 5 is not visible)
- Select the one for which you'd like the data aggregated

z/OS Data Gatherer Connection Details

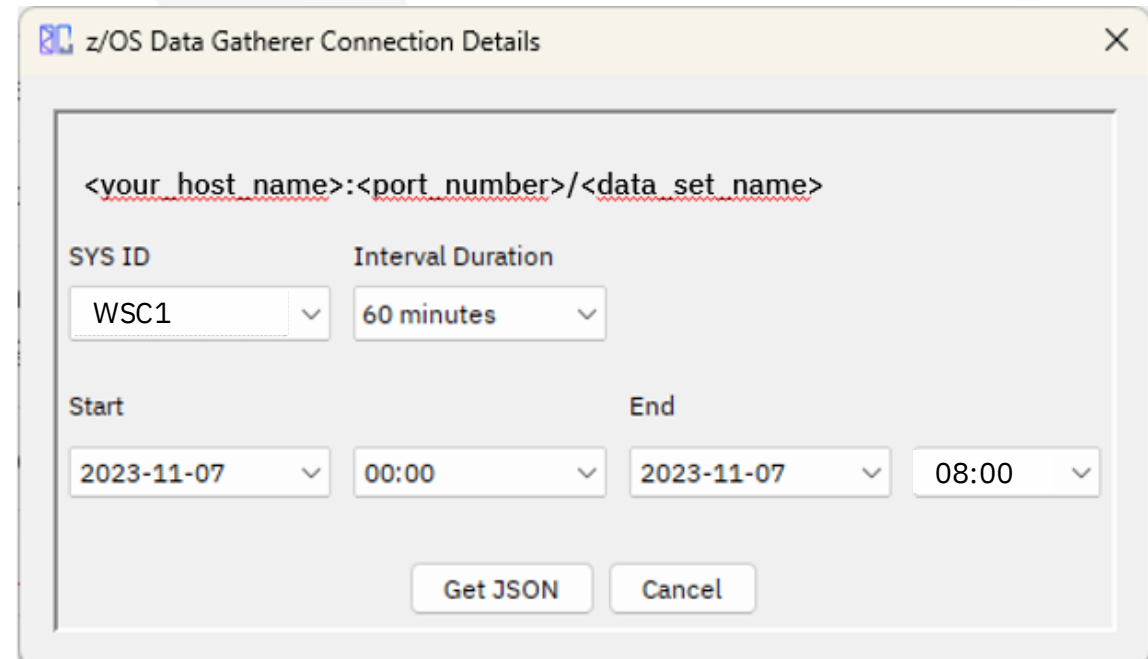
<your host name>:<port number>/<data set name>

SYS ID	Interval Duration	Start	End
WSC1	60 minutes	2023-11-07	2023-11-07 23:00

Get JSON Cancel

Providing More Parameters Time Span

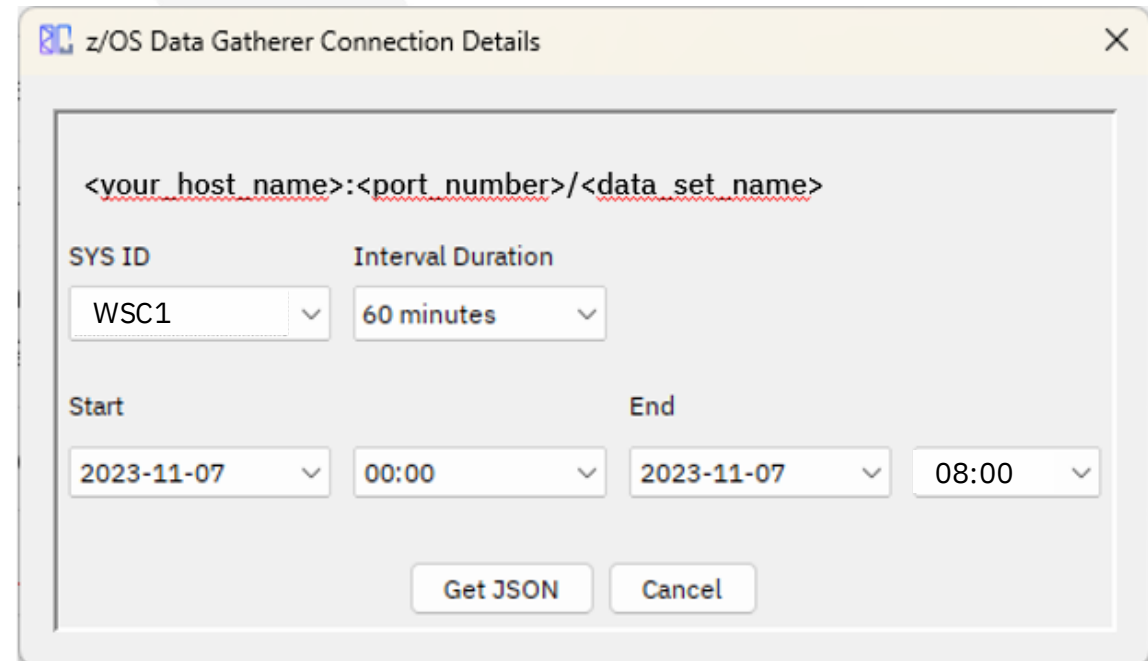
- All available dates and times in the SMF are presented
- Limit the data span using the Date and Time selectors



The screenshot shows a dialog box titled "z/OS Data Gatherer Connection Details". It contains a text field with the placeholder text "<your host name>:<port number>/<data set name>". Below this are two rows of dropdown menus. The first row is labeled "SYS ID" and "Interval Duration", with "WSC1" and "60 minutes" selected. The second row is labeled "Start" and "End", with "2023-11-07" and "08:00" selected. At the bottom are two buttons: "Get JSON" and "Cancel".

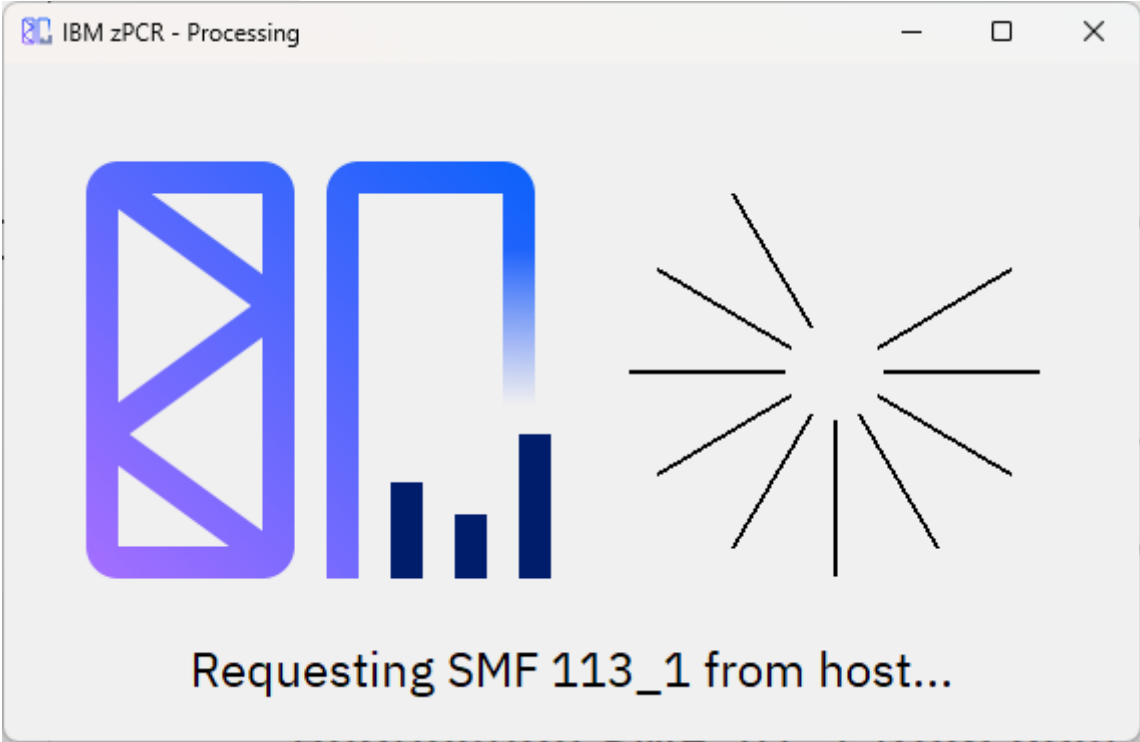
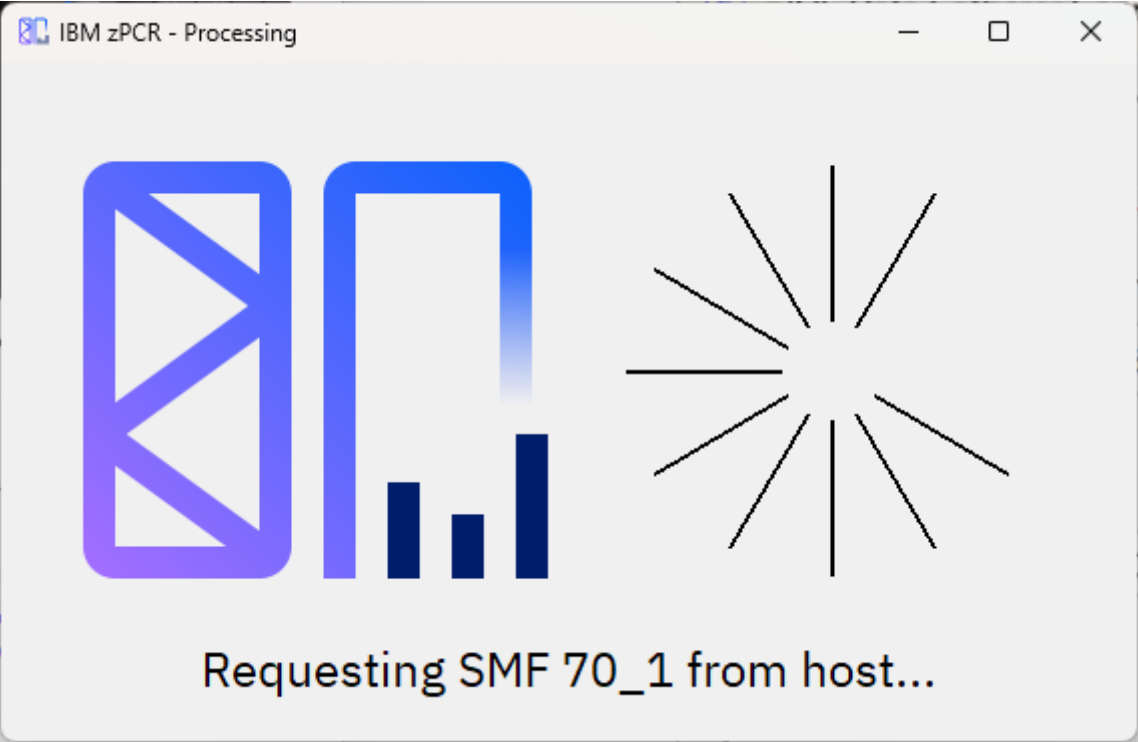
Select zip file name and destination

- Click “Get JSON”
- Name the zip file to be saved and select its destination




The screenshot shows a dialog box titled "z/OS Data Gatherer Connection Details". It contains a text input field with the placeholder text "<your host name>:<port number>/<data set name>". Below this are two rows of dropdown menus. The first row is labeled "SYS ID" and "Interval Duration", with "WSC1" and "60 minutes" selected respectively. The second row is labeled "Start" and "End", with "2023-11-07 00:00" and "2023-11-07 08:00" selected. At the bottom of the dialog are two buttons: "Get JSON" and "Cancel".

Wait for data



Interval Selection Window

JSON Interval Selection IBM zPCR 9.7.4.json

z/OS Data Gatherer JSON Intervals
#1  Configuration #1

JSON Zip File Name: C:\Users\ShawnLundvall\Documents\zPCR Defaults\JSON Zip Files\demo_SHARE.zip

Relative Interval Number	CPC ID	GP Processor Model	Date	Time	Interval Length	Number of Active Partitions	Includes CPU-MF	Topology Changed	Pool 1 GP Pool Utilization
1.	CPCWSCZ1	3931-7E7	2023-11-07	00:00:00	01:00:00	8	✓		22.13%
2.	CPCWSCZ1	3931-7E7	2023-11-07	01:00:00	01:00:00	8	✓		26.26%
3.	CPCWSCZ1	3931-7E7	2023-11-07	02:00:00	01:00:00	8	✓		16.83%
4.	CPCWSCZ1	3931-7E7	2023-11-07	03:00:00	01:00:00	8	✓		16.71%
5.	CPCWSCZ1	3931-7E7	2023-11-07	04:00:00	01:00:00	8	✓		21.40%
6.	CPCWSCZ1	3931-7E7	2023-11-07	05:00:00	01:00:00	8	✓		21.99%
7.	CPCWSCZ1	3931-7E7	2023-11-07	06:00:00	01:00:00	8	✓		25.31%

Table View

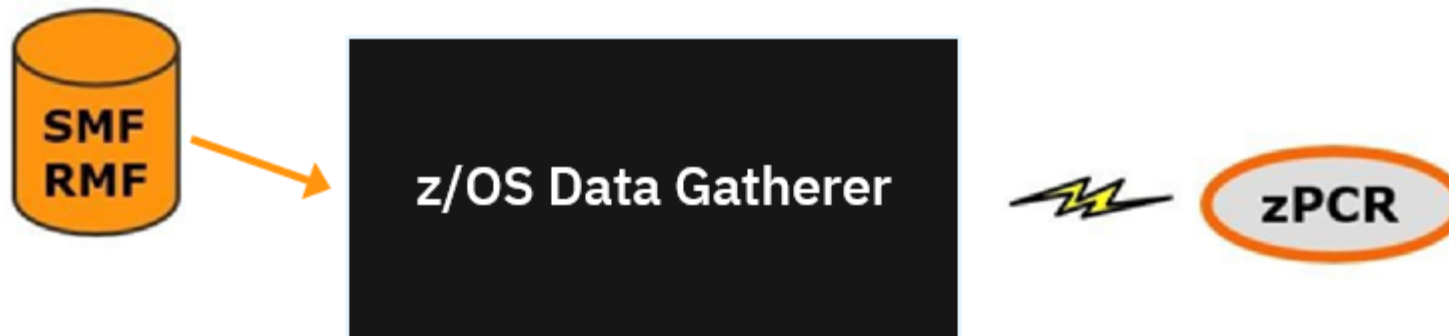
Show All Pools Number of intervals: 23

Get JSON Load JSON Show Partitions

Click on a row to select interval for which zPCR partition definitions are to be created

Using z/OS Data Gatherer REST Services to provide SMF & RMF data to IBM zPCR modernizes your workflow.

- z/OS Data Gatherer REST Services make data on the host available via REST calls and present it in JSON
- IBM zPCR loads JSON data directly



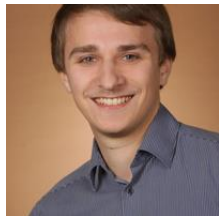
- Pros: Simpler, no host interaction required
- Cons: Longer runtime for data extraction

Feedback? Email
zpcr@us.ibm.com

IBM Z Trial for z/OS Data Gatherer and SMF Explorer:

<https://early-access.ibm.com/software/support/trial/cst/welcomepage.wss?siteId=1948&tabId=5393&w=1>

Get in contact with us:



Alexander Giemsa
z/OS Data Gatherer

Alexander.Giemsa@ibm.com



IBM WSC zPCR Team

zPCR@us.ibm.com

Session Tomorrow:

**Cracking the Code to z17
Capacity Sizing – BYOD Lab
Tuesday, 8am, STE Lab**

Experience more with IBM



Visit us at the IBM Booth #113

After a full day of technical sessions, take a break with us!

Connect with our experts, snap a photo with the z17 Plexi or the latest Telum II, and get an up-close look at our Spyre Accelerator.

Come back each day for fresh topics and demos at our expert stations.

Think 2026

Join 5000+ senior business and technology leaders who are seizing the AI revolution to unlock unprecedented growth and productivity at **Think 2026**.

Find out more information using the QR code below.



IBM Digital Asset Haven

IBM Digital Asset Haven is the operational backbone for financial institutions and regulated enterprises entering the digital asset economy.

Find out more information using the QR code below.



Your feedback is important!

Submit a session evaluation for each session you attend:

www.share.org/evaluation

Modernize Your Data Sources: An IBM zPCR Story

