

# **NonStop monitoring and automation**

**Wolfgang Breidbach**

## Bank-Verlag

- Founded in 1961 as the publishing house of the magazine „Die Bank“.
- Running on IBM Systems /1 and /370 the first Authorisation Center in Germany for ATM-transactions was founded at the Bank-Verlag in 1986.
- In 1988 authorisation was migrated to Tandem creating the first active-active application.
- In the following years we took our way through Cyclone, CLX, CLX2000, K10000, K20000, S7000, S70000, S72000 to at last S86000
- 2005 we moved to Integrity NonStop
- 2010 the secondary datacentre was moved to a new location
- 2012 we migrated our production systems to NonStop blades
- Today we are **the** IT-service provider for the Private Banks in Germany



---

## The start

- Bank-Verlag was using a commercial monitoring tool
- Management decided to replace that tool by open source Nagios for all Windows, Unix and Linux systems
- Nagios should be used for NonStop systems as well
- Problem: No open source monitoring tool for NonStop available that fulfilled our needs
- Decision: We will have to create something ourselves!

---

## Some basic decisions

- The main purpose is monitoring our NonStop systems
- Feeding Nagios with information should be a result of that
- The open source world is changing quickly, we should be able to support any other tool with little changes
- The NonStop monitoring should not depend on any external tool
- The messages should not require in-depth NonStop knowledge
- Avoid manual configuration wherever possible



## Our approach

- We have a bunch of „subsystems“ like CPU, Pathway, Lines, NetBatch and so on
- Every subsystem has its own monitoring module
- Every module collects all available configuration information automatically like
  - NetBatch module collects all information concerning NetBatch jobs and calendars
  - Line Module collects all lines
- Some modules need additional configuration data:
  - File module needs the filesets to check
  - EMS module needs the messages to look for



## Our approach

- Every module has a „refresh configuration“ function
- Every module is configurable with parameters, every parameter has a default
- If an event is found that could be handled by the toolbox it should be handled by the toolbox
  - File is getting full => perform a reload or increase maxextents
  - A static Pathway server is down => issue a START command
  - A process is consuming too many CPU cycles => reduce priority



---

## Our approach

- Another goal was avoiding manual tasks we do not like
  - Regular reloads
  - Checking Backups
  - Checking database contents
- Collect statistical data
  - Line usage
  - File sizes
  - CPU usage
  - TMF rate
- Create documentation about the configuration of the system



## Our approach

- We want to make information available to people not familiar with NonStop systems
  - The X.25 line with the calling address 12345678 is connected to the SWAN-box with the „S77“ sticker on Clip 1 line 0
  - The TCP/IP connection with the address 192.168.77.77 is configured on the controller in slot 2.4 on „D“ and the port has the MAC address 08.00.12.34.56
  - This should be database information accessible and usable without any detailed NonStop knowledge
- Reports of installed hardware should be understandable without the knowledge of HP product numbers





---

## The Start

- First subsystem was „CPU and processes“
- Development based on some already available programs
- The CPU- and processmonitoring program should not write any diskfiles
- Create the tools to maintain the appropriate tables including the long-term data collection
- Create a central message collector reading the tables and formatting the messages
- Continue with the other subsystems

---

## The next steps

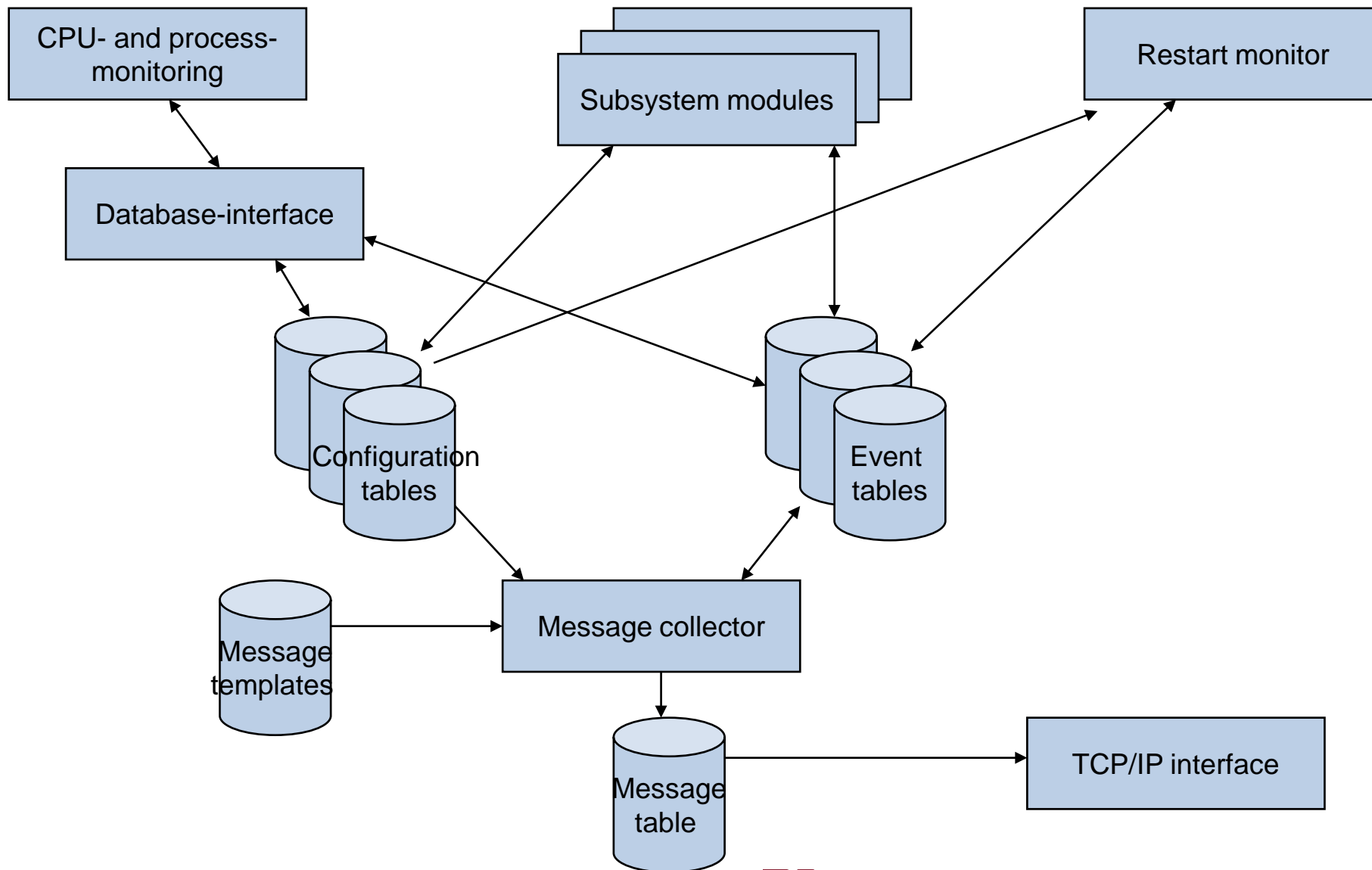
- Decision to build the software like a product
- Great advantages distributing the software on our 4 (at the moment 6) systems
- Design of a central message handling program
- Avoid any hard-coded messages
- A side-effect: The toolbox supports multiple languages

## Available subsystems

- CPU- and Processes (incl. automatic restart of processes \*)
- Lines
- Pathway
- Files incl. automatic reload \*
- TMF
- RDF
- Netbatch
- Devices
- TCP/IP
- Spooler
- EMS-messages \*
- Message collector
- Backups \*

\* = configuration required





## Some additional information

- The original monitoring toolbox is based on SQL tables
- An Enscribe version is in progress
- The toolbox is not depending on Measure, Measure is only used to find the originator of a heavy diskload
- The toolbox is causing very little CPU-load,
- Collected statistical data allows lots of reports using standard tools like Excel

## Advantages

- Keep track of hardware changes like exchange of disks
- No need for additional software like Measure
- Software is running „out of the box“ without a need for additional configuration
- Lots of parameters and table entries for configuration available
- The software supports multiple languages, at the moment the messages are available in German and English
- Bank-Verlag is not a vendor but a user, we are using the software ourselves
- Very limited commercial interest in selling the software



---

## Advantages during daily life

- Reloads are carried out automatically if needed
- Processes causing heavy diskload are found (Measure required!)
- The priority of processes using too many CPU cycles can be automatically reduced
- Pathway-servers can be automatically restarted
- Missing processes can be restarted automatically
  - Existence of required processes can be checked
  - The whole system including all the applications can be started this way!

---

## Advantages during daily life

- Batchjobs and Calendars are checked periodically.
- If a calendar is expiring, a message is issued a few days before expiration
- The outcome of all backup jobs is checked
- Disk problems are checked periodically including
  - Number of ZZSA files
  - Status of OSS-filesets



---

## Advantages during daily life

- Files matching predefined filesets are checked for files running full
- If a file is too full it is automatically checked for a possible reload or the maxextents are increased
- All configured files are periodically reloaded if necessary
- Necessary reload is decided depending on slack and fragmentation
- All needed parameters can be defined globally, for a fileset or even for a single file.
- The need for manual reloads has been reduced to zero

## Interesting problems

- The status of TCP/IP connections can be checked
  - You need 2 established connections from your \$ZB000 (192.168.77.77) to 192.168.88.88 port 1234.
  - If at least one of these connections is down, a message is created
  - The cause for that might be an erroneously changed firewall configuration
- The same feature has been implemented for X.25 connections

## A real life case concerning TCP/IP

- Our NonStop is accessing another server through a firewall
- There have to be 2 established connections on port 4711
- A rule within the firewall was erroneously changed
- The NonStop could no longer establish a new connection to the server
- The already established connections were not affected
- The real problem we had weeks later when one of the connections had to be reestablished
- The monitoring tool found the missing connection immediately



---

## Another problem

- We have a leased line to another provider
- Line is using X.25 protocol
- During peak hours we had some problems on the line
- Using the statistical data we found out that the capacity of the line was exceeded
- Increasing the speed immediately solved all problems

## Security issues

- Safeguard reports erroneous logons
- Safeguard does not report the external origin of this logon like the IP-address
- We read the Safeguard log and add that information
- So the question „From where did the logon with Administrator to the NonStop come“ can be answered by a look at our table

## Application monitoring

- There are 2 kinds of application monitoring:
  - Checking database contents
  - Checking application messages
- The database contents are checked using SQL-statements of the type „SELECT COUNT(\*) from ... WHERE... BROWSE ACCESS;“
- The result is compared against given values and a message is created if necessary
- The severity of the messages can be set depending on the result like:
  - 1 found => Warning
  - 2 found => Error



## Checking EMS-messages

- Our applications are using EMS collectors to report any errors
- We are able to check the number of messages per type per time period
- A sample message would be „Timeout process \$ABCD“, process \$ABCD is routing messages to XY-Bank
- We define the message be „Timeout“ and „\$ABCD“ as „Timeout to XY-BANK“ and count those messages per period
- A messages is created depending on the configured theshold for this type of message

## An idea for EMS message handling

- We are handling authorisation requests for credit and debit cards, most of these requests are send to the card-issuing banks
- We are creating minute-based statistics of those requests per issuer
- If an issuer has problems we can create a message like  
60% of the requests unsuccessful
- Now the message handling gets this information and handles it according to the configuration:
  - 1 message within 10 minutes → no need for action
  - 10 messages within 10 minutes → create an alarm






# Our main Nagios screen for NonStop

AZ Übersicht




# Our main Nagios screen for NonStop with error message

## AZ Übersicht

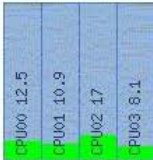
BVPROD0 

Leitungen

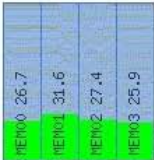
Leitungen 

BVPROD1

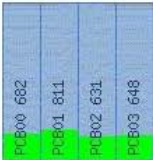
CPU



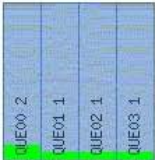
MEM




PCB



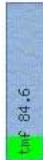
QUE




TMF




TMF



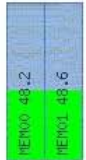
QUE




PCB



MEM



CPU



System	
	PATHWAY
	FILES
	PROCESS
	TMF

Dep			
	001		013
	002		014
	003		015
	004		016

AC/SC	
	AC 50
	AC 100
	SC 50
	SC 100

AC/SC	
	AC 50
	AC 100
	SC 50
	SC 100
	TRISTAN 50
	TRISTAN 100

Dep			
	001		013
	002		014
	003		015
	004		016
	005		017
	006		018
	007		019
	008		020
	009		021
	010		022
	011		023
	012		024

System	
	PATHWAY
	FILES
	PROCESS
	TMF
	TCPIP
	RDF
	BAT
	DEVICE
	SPOOL
	EMS

Konfig	
	Release

Service (Last state refresh: 2012-09-24 12:29:00)

Host Name: BVPROD0 (BVPROD0)

Service Name: TMF Message

Summary State: **CRITICAL**

Summary Output: CRIT - C 3028 "Audittrail \$TMF.ZTMFAT.AA049409 nicht freigegeben, wartet auf RDF"

Last Check: 2012-09-24 12:29:00

Next Check: 2012-09-24 10:22:41

Last State Change: 2012-09-24 10:21:41

PNP-Graph



---

**Any questions???**

**Wolfgang Breidbach**

Bank-Verlag GmbH

IT-Services

Wendelinstr. 1

50933 Köln

E-Mail: [Wolfgang.Breidbach@Bank-Verlag.de](mailto:Wolfgang.Breidbach@Bank-Verlag.de)

[www.Bank-Verlag.de](http://www.Bank-Verlag.de)

