

Начальнику Управления информационно-телекоммуникационной инфраструктуры
ФГБУ ЦИТО
Швецову Д.И.

Аналитическая записка

В настоящей аналитической записке приводятся результаты анализа и предлагаются пути решения проблемы потери документов в системе Межведомственного электронного документооборота для служебного пользования (далее МЭДО ДСП). Суть проблемы: потеря документов при их обработке в системе МЭДО ДСП на узлах 192.168.162.20 (узел «medo-dsp»), 192.168.162.52 (узел «app1»), 192.168.152.37 и 192.168.162.57.

Граф связности узлов на третьем (двунаправленные связи, синий цвет) и седьмом (двунаправленные связи, зеленый цвет) уровнях OSI, вместе с направлениями инициации соединений (однаправленные связи от инициатора к цели) представлен на рисунке 1.

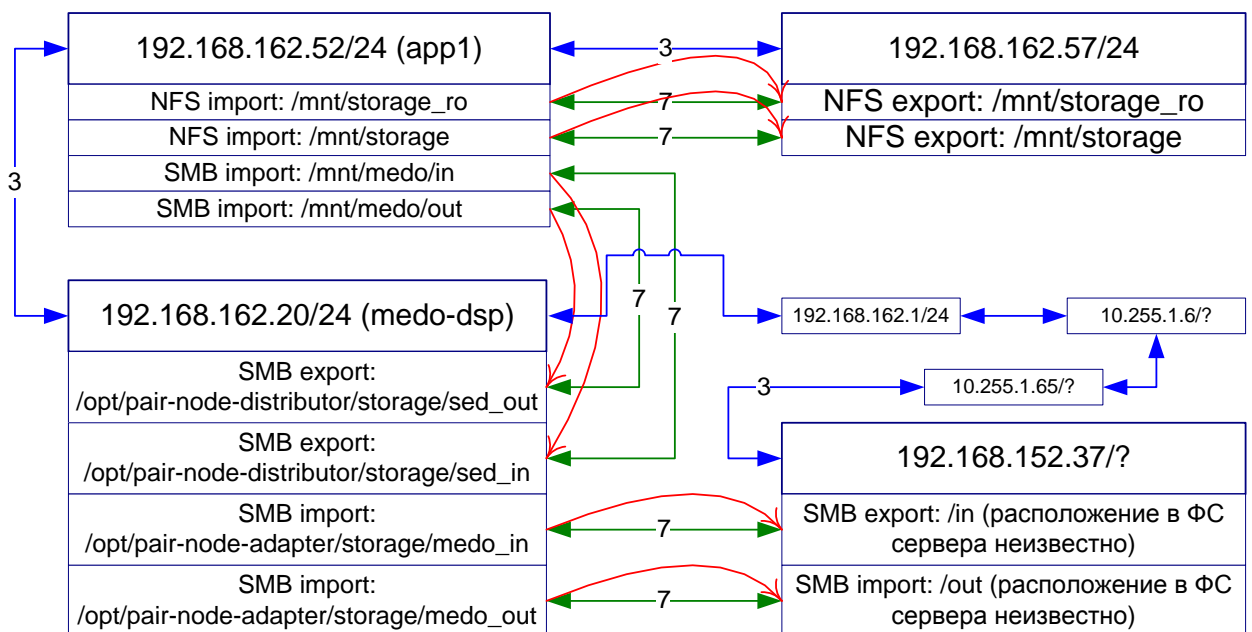


Рисунок 1 – Граф связности узлов на транспортном и прикладном уровнях

Значения доступного пространства на экспортируемых ресурсах (по состоянию на 24.05.2024) сведены в таблицу 1.

Таблица 1 – доступное пространство на экспортируемых ресурсах

Ресурс	Всего	Доступно	Исп., %
/mnt/storage_ro на 192.168.162.57	4,0 Тб	151 Гб	97
/mnt/storage на 192.168.162.57	8,0 Тб	2,1 Тб	75
/opt/pair-node-distributor/storage/sed_out на 192.168.162.20	1 Тб ¹	900 Гб ²	15
/opt/pair-node-distributor/storage/sed_in на 192.168.162.20	1 Тб ¹	900 Гб ²	15
/in на 192.168.152.37	90 Гб ³	35 Гб	73
/out на 192.168.152.37	90 Гб ³	35 Гб	73

Примечания:

¹ – Ресурс расположен непосредственно в корневой файловой системе.

² – Значение доступного пространства по показаниям ОС сервера 192.168.162.20 (df -h | grep "/\$") не совпадает со значением по показаниям ОС клиента 192.168.162.52 (df -h | grep "162\.20").

³ – Возможно, ресурс расположен непосредственно в корневой файловой системе.

Как следует из таблицы 1, для всех экспортируемых ресурсов (за исключением ресурса `storage_ro` на узле 192.168.162.57, смонтированного на узле «app1» без доступа на запись) процент использованного пространства не приближается к критическим значениям.

Анализ проблемы был выполнен сотрудниками Управления информационно-телекоммуникационной инфраструктуры (УИТКИ) в апреле 2024 г. Установлено, что потеря документов соответствует моменту возрастания числа подкаталогов первого уровня в каталоге `/mnt/medo_out` на узле «app1» (фактически, это каталог `/opt/pair-node-distributor/storage/sed_out` на узле «medo-dsp»). Сотрудниками УИТКИ средствами `zabbix` был настроен мониторинг каталога `/opt/pair-node-distributor/storage/sed_out`. На основании результатов мониторинга при срабатывании триггера по числу подкаталогов оператор выполняет перезапуск службы `pair-node-distributor`. Дополнительно выполняется мониторинг каталога `/opt/pair-node-distributor/storage/sed_in`; срабатывание триггера по числу подкаталогов для этого каталога было зарегистрировано, в частности, 24.05.2024.

Возможны по меньшей мере три варианта решения проблемы, не требующие периодического внимания операторов из числа сотрудников УИТКИ.

1. Анализ исходного кода службы `pair-node-distributor`, локализация проблемы и ее решение разработчиком службы `pair-node-distributor`.

2. Анализ исходного кода службы `pair-node-distributor`, локализация проблемы и ее решение сотрудниками УИТКИ.

3. Маскировка проблемы, обеспечивающая в нулевом приближении ее решение силами сотрудников УИТКИ.

Второй вариант не может быть рекомендован по нескольким причинам:

– общее число исходных файлов (*.php) службы (для версии 3.1.8.4120) составляет 170, число строк – 24702;

– анализ и корректировку кода службы нужно будет выполнять после каждого ее обновления;

– анализ и корректировка кода службы, разработанной третьей стороной, не входит в область ответственности сотрудников УИТКИ;

– анализ и корректировка кода службы, оперирующей документами ДСП, сотрудниками УИТКИ недопустима.

Первый вариант является оптимальным.

В настоящей аналитической записке раскрывается третий вариант, состоящий в создании дополнительной службы мониторинга и перезапуска сервиса `pair-node-distributor`.

Служба включает:

– сценарий на языке Bourne shell, который на узле «medo-dsp» должен быть размещается в каталоге, входящем в переменную среды PATH суперпользователя (предпочтительно – в `/opt/bin` или `/usr/local/bin`);

– конфигурационный файл, имя которого совпадает с именем сценария, имеющий дополнительно расширение «`conf`».

Поиск конфигурационного файла выполняется в следующем порядке:

– каталог `etc` на уровень выше каталога, в котором размещен сценарий;

– каталог `/opt/etc`;

– каталог `/usr/local/etc`;

– каталог, в котором размещен сценарий.

В конфигурационном файле должны быть установлены следующие переменные:

– `MON1_DIR`: первый из каталогов, подлежащих мониторингу;

– `MON2_DIR`: второй из каталогов, подлежащих мониторингу;

- WARN_NO: число подкаталогов первого уровня, при достижении которого выдается предупреждающее сообщение;
- CRIT_NO: число подкаталогов первого уровня, при достижении которого выдается сообщение об ошибке и выполняется заданное действие;
- CRIT_ACTION: действие, выполняющееся при достижении критического числа подкаталогов.

При снижении числа подкаталогов ниже значения WARN_NO (при условии, что предыдущее значение было не ниже WARN_NO) выдается сообщение о штатном состоянии.

В переменной CRIT_ACTION могут быть использованы допустимые в системе команды оболочки, в т.ч. фильтры и конвейеры.

Дополнительно в конфигурационном файле могут быть установлены переменные:

- TRIGGER1_FILE: имя файла (абсолютный путь) состояния первого каталога мониторинга (по умолчанию «/tmp/ndmon.trigger1»);
- TRIGGER2_FILE: имя файла (абсолютный путь) состояния второго каталога мониторинга (по умолчанию «/tmp/ndmon.trigger2»);
- TO_CRIT_FROM_OTHER_MSG: текст сообщения об ошибке (по умолчанию ««имя_скрипта-год-месяц-день»: critical number “значение CRIT_NO” of TL subdirs, executing действие “значение CRIT_ACTION”»);
- TO_WARN_FROM_NORM_MSG: текст предупреждения (по умолчанию ««имя_скрипта-год-месяц-день»: warning: number of TL subdirs reached “значение WARN_NO”»);
- TO_NORM_MSG: текст сообщения о переходе в штатное состояние (по умолчанию ««имя_скрипта-год-месяц-день»: number of TL subdirs returned to normal (less then “значение WARN_NO”»).

Если конфигурационный файл не найден, используются значения по умолчанию (в т.ч. для переменных MON1_DIR, MON2_DIR, WARN_NO, CRIT_NO и CRIT_ACTION, см. исходный код сценария).

Листинг сценария приведен в приложении 1 и доступен по URL:

<https://cito.os22.ru/files/nddoublemon/opt/bin/nddoublemon>

Пример конфигурационного файла приведен в приложении 2 и доступен по URL:

<https://cito.os22.ru/files/nddoublemon/opt/etc/nddoublemon.conf>

Установка сценария включает копирование его в требуемый каталог, создание (в т.ч. на основе приложения 2) конфигурационного файла и обеспечение его периодического вызова средствами ОС. Для ОС с функционирующей службой cron (или ее эмуляцией, в т.ч. средствами systemd) достаточно добавить в файл /etc/crontab соответствующую строку. Например, для периодического запуска с интервалом в одну минуту:

```
* /1 * * * * root /opt/bin/nddoublemon
```

Получение оповещений на электронную почту потребует настройки sendmail-подобного агента передачи почты (SMTP-клиента; проверка выполнялась для МТА msmtп, рисунок 2).





Сегодня			
	silentrap@gmail.com	Cron <root@web2> /opt/bin/nddoublemon nddoublemon-2024-05-27: number of TL subdirs returned to normal (less then 3)	2:53
	silentrap@gmail.com	Cron <root@web2> /opt/bin/nddoublemon nddoublemon-2024-05-27: critical number 4 of TL subdirs! Executing "echo Muhaha ! mail..."	2:47
	silentrap@gmail.com	nddoublemon_DEBUG Muhaha !	2:47
	silentrap@gmail.com	Cron <root@web2> /opt/bin/nddoublemon nddoublemon-2024-05-27: warning: number of TL subdirs reached 3.	2:44

Рисунок 2 – Пример сообщений о срабатывании триггеров

Для ОС, основанных на systemd и не включающих ни реализацию, ни эмуляцию службы cron, потребуется создание таймер-модуля systemd.

Приложение 1. Листинг сценария

```
#!/bin/sh

# Counts number _nd of top-level subdirs in two MON_DIRS and performs
# actions (if not empty) in case of:
# a) _nd reached level CRIT_NO, previous level was not CRIT_NO;
# b) _nd reached level WARN_NO, previous level was normal;
# c) _nd returned to normal level, previous level was not normal.

die() {
    echo $@
    exit 1
}

# These variables should be altered in config file.
# Absolute path to directories to monitor
MON1_DIR="/tmp/mon1"
MON2_DIR="/tmp/mon2"
# Number of TD subdirs: warning level.
WARN_NO=2
# Number of TD subdirs: critical level.
CRIT_NO=3
# Action to execute in case a). Any shell command (s), pipes allowed, but no local vars.
CRIT_ACTION="echo Muhaha ! | mail -s nddoublemon_DEBUG root"

# "Semi-local" variables (for use in config also).
_me="$(basename $0)"
_date="$(date +%Y-%m-%d)"

# These variables may be altered in config file.
# Absolute paths to trigger files
TRIGGER1_FILE="/tmp/ndmon.trigger1"
TRIGGER2_FILE="/tmp/ndmon.trigger2"
# Messages.
# Note: ${_me} and ${_date} are valid at the moment and can be used in config.
# Note: if WARN_NO, CRIT_NO and CRIT_ACTION are to be set in config, they should set be above
messages.
TO_CRIT_FROM_OTHER_MSG="${_me}-${_date}: critical number ${CRIT_NO} of TL subdirs, executing
\"${CRIT_ACTION}\"."
TO_WARN_FROM_NORM_MSG="${_me}-${_date}: warning: number of TL subdirs reached ${WARN_NO}."
TO_NORM_MSG="${_me}-${_date}: number of TL subdirs returned to normal (less then ${WARN_NO})"

# Source config file
_config_file="$(dirname $0)/../etc/${_me}.conf"
[ -r "${_config_file}" ] || _config_file="/opt/etc/${_me}.conf"
[ -r "${_config_file}" ] || _config_file="/usr/local/etc/${_me}.conf"
[ -r "${_config_file}" ] || _config_file="$(dirname $0)/${_me}.conf"
[ -r "${_config_file}" ] && . "${_config_file}" \
# || echo "${_me}-${_date}: warning: unable to find configuration file ${_config_file}."
# If you need a lot of warnings, just uncomment the line above.

# These variables are for internal use only
_norm_id="NORM"
_warn_id="WARN"
_crit_id="CRIT"

# If there is no saved state, assume it was NORM.
[ -f "${TRIGGER1_FILE}" ] || echo ${_norm_id} > "${TRIGGER1_FILE}"
[ -f "${TRIGGER2_FILE}" ] || echo ${_norm_id} > "${TRIGGER2_FILE}"
[ -r "${TRIGGER1_FILE}" -a -r "${TRIGGER2_FILE}" ] || \
die "${_me}-${_date}: unable to create trigger file(s) \"${TRIGGER1_FILE}\" and/or
\"${TRIGGER2_FILE}\"."

# This function should only be called from check_dir_for_trigger()
# arg no. 1: MON_DIR, we assume it exists.
determine_state() {
    _mon_dir="$1"
    _nd=$(find "${_mon_dir}" -mindepth 1 -maxdepth 1 -type d | wc -l)
    [ ${_nd} -lt ${WARN_NO} ] && echo ${_norm_id} && exit
    [ ${_nd} -lt ${CRIT_NO} ] && echo ${_warn_id} && exit
    echo ${_crit_id}
}

# Actual work.
# arg no. 1: MON_DIR, will be checked for existance.
# arg no. 2: TRIGGER_FILE, will be checked for rw access.
check_dir_for_trigger()
```

```

{
    _mon_dir="$1"
    _trigger_file="$2"
    _caller="check_dir_for_trigger()"
    [ -d "${_mon_dir}" ] || \
        die "${_me}-${_date}: ${_caller}: directory \"${_mon_dir}\" does not exist, check
config \"${_config_file}\"."
    [ -r "${_trigger_file}" ] || \
        die "${_me}-${_date}: ${_caller}: trigger \"${_trigger_file}\" is unreadable,
check config \"${_config_file}\"."
    [ -w "${_trigger_file}" ] || \
        die "${_me}-${_date}: ${_caller}: trigger \"${_trigger_file}\" is unwritable,
check config \"${_config_file}\"."
    _old_state="$(cat "${_trigger_file}")"
    _new_state="$(determine_state "${_mon_dir}")"
    if [ "${_new_state}" = "${_crit_id}" -a "${_old_state}" != "${_crit_id}" ]; then
        # Case a)
        [ -n "${TO_CRIT_FROM_OTHER_MSG}" ] && echo "${TO_CRIT_FROM_OTHER_MSG}"
        if [ -n "${CRIT_ACTION}" ]; then
            eval "${CRIT_ACTION}"
        fi
        echo "${_new_state}" > "${_trigger_file}"
        return 1
    fi
    if [ "${_new_state}" = "${_warn_id}" -a "${_old_state}" = "${_norm_id}" ]; then
        # Case b)
        [ -n "${TO_WARN_FROM_NORM_MSG}" ] && echo "${TO_WARN_FROM_NORM_MSG}"
    elif [ "${_new_state}" = "${_norm_id}" -a "${_old_state}" != "${_norm_id}" ]; then
        # Case c)
        [ -n "${TO_NORM_MSG}" ] && echo "${TO_NORM_MSG}"
    fi
    echo "${_new_state}" > "${_trigger_file}"
    return 0
}

# main()
check_dir_for_trigger "${MON1_DIR}" "${TRIGGER1_FILE}"
_retval1=$?
check_dir_for_trigger "${MON2_DIR}" "${TRIGGER2_FILE}"
_retval2=$?
[ "${_retval1}" -eq 0 -a "${_retval2}" -eq 0 ] && exit 0 || exit 1

```

Приложение 2. Пример конфигурационного файла.

```
MON1_DIR="/opt/pair-node-distributor/storage/sed_out"
MON2_DIR="/opt/pair-node-distributor/storage/sed_in"

WARN_NO=100
# WARN_NO=3
# Number of TD subdirs: critical level.
CRIT_NO=190
# Action to execute in case a). Any shell command (s), pipes allowed, but no local vars.
CRIT_ACTION="systemctl restart pair-node-distributor"

# Note: ${_me} and ${_date} are valid since we are sourced after their initialization.
# Note: if WARN_NO, CRIT_NO and CRIT_ACTION should set be above or not at all.
TO_CRIT_FROM_OTHER_MSG="${_me}-${_date}: critical number ${CRIT_NO} of TL subdirs! Executing
\"${CRIT_ACTION}\"."
TO_WARN_FROM_NORM_MSG=""
TO_NORM_MSG="${_me}-${_date}: number of TL subdirs returned to normal (less then ${WARN_NO})"
```