

## РАСПРЕДЕЛЕННАЯ СИСТЕМА РЕЗЕРВНОГО КОПИРОВАНИЯ ДЛЯ МАЛЫХ ПРЕДПРИЯТИЙ<sup>1</sup>

**Кузнецов Л.К.,**  
кандидат технических наук, доцент,  
ФГБОУ ВПО «Финансовый университет  
при Правительстве российской Федерации», Москва,  
longin\_kuz@mail.ru

**Аннотация.** Рассматриваются вопросы резервного копирования данных для малых предприятий. Преимуществом предлагаемой системы является отсутствие необходимости в выделенном сервере.

**Ключевые слова:** резервное копирование, данные, жесткий диск, сервер, программа.

## DISTRIBUTED SYSTEMS BACKUP FOR SMALL ENTERPRISES

**Kuznetsov L.K.,**  
Ph.D., Associate Professor,  
FGOBU VPO "Financial University under the Government  
of the Russian Federation", Moscow

**Abstract.** Addresses issues of data backup for small businesses. The advantage of the proposed system is the lack of not-need a dedicated server.

**Key words:** backup, data, hard disk, server, software

**Р**егулярное резервное копирование данных является неотъемлемой частью успешного бизнеса. Потеря информации и невозможность быстрого ее восстановления может привести к огромным убыткам, вплоть до полного прекращения деятельности компании [1]. К сожалению, зачастую о проблеме начинают задумываться слишком поздно. Ситуация усугубляется в небольших компаниях, где может не хватать ресурсов для привлечения квалифицированных специалистов, способных качественно и надежно организовать резервное копирование.

В данной работе рассмотрен один из возможных подходов к решению проблемы. Хотя и существует ряд ограничений, препятствующих его использованию в крупных компаниях, у него есть ряд преимуществ, делающих его актуальным для небольших компаний. К преимуществам распределенной системы резервного копирования, предлагаемой в данной работе, можно отнести простоту внедрения,

отсутствие необходимости в выделенном сервере, возможность задействовать имеющиеся ресурсы (неиспользуемое дисковое пространство) оптимальным образом.

Разработка собственного решения, несомненно, повлечет дополнительные расходы, но желание полностью использовать имеющиеся ресурсы, наличие множества технологий, позволяющих сильно сократить сложность разработки и затраченное время, а также потенциальная возможность продажи разработанного продукта аналогичным малым предприятиям, послужили основанием для принятия решения в его пользу.

К разрабатываемой системе были предъявлены следующие функциональные требования:

- работа по расписанию – резервное копирование должно начинаться автоматически в строго установленное время, это поможет минимизировать ошибки связанные с человеческим фактором (забывчивость человека);
- возможность разбиения резервной копии на части с добавлением избыточной информации

<sup>1</sup> По материалам «II Международного конгресса по информационной безопасности национальных экономик INFOSECURITYFINANCE»

– файл разбивается на  $n$  частей, дополнительная избыточная информация создается путем генерации еще  $m$  частей таким образом, что из любых  $n$  частей можно будет восстановить исходный файл, это позволит увеличить надежность системы;

- использование одноранговой  $p2p$  сети – после разбиения все части файла распределяются по сети, система должна обеспечивать их передачу, а также прием и сохранение;
- восстановление резервных копий – в случае отказа оборудования и утери данных, по запросу пользователя, система должна предоставить ему список резервных копий, доступных для восстановления, переслать части и восстановить из них исходный файл;
- гибкость настройки – параметры системы должны быть изменяемы, например, путь к файлам, резервные копии которых будут создаваться, путь, по которому сохраняются принятые части файлов, время начала работы, уровень избыточности;
- ведение журнала работы – все действия необходимо записывать в специальные файлы журналов, что поможет выявить неполадки в работе системы, оценить скорость проведения операций, принять верные решения, связанные с оптимизацией работы системы.

В качестве архитектуры системы выбрана одноранговая сеть  $p2p$ .  $P2p$  (peer-to-peer) – компьютерная сеть, основанная на равноправии участников. В такой сети отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и сервером. В отличие от архитектуры клиент-сервер, такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов [3]. Таким образом, разрабатываемая система является полностью распределенной и не требует выделенного сервера.

Для надежного распределения резервных копий по узлам одноранговой сети, необходимо ввести избыточную информацию, то есть сгенерировать

дополнительные части резервной копии. Для решения этой задачи подходит алгоритм, используемый в RAID (Redundant Arrays of Inexpensive Disks) массивах [1]. Работа данного алгоритма основана на корректирующих кодах Рида-Соломона.

Рассмотрим алгоритм [4]. Перед нами стоит следующая задача: пусть имеется  $n$  частей резервной копии размером  $b$  байт каждая  $D_1, D_2, \dots, D_n$ , необходимо сгенерировать еще  $m$  частей размером  $b$  байт каждая  $C_1, C_2, \dots, C_m$  таким образом, что если будут утеряны любые  $m$  частей из  $D_1, D_2, \dots, D_n, C_1, C_2, \dots, C_m$ , резервная копия может быть восстановлена.

Будем рассматривать каждую часть как последовательность слов. Размер каждого слова составляет  $w$  бит и выбирается программистом (с учетом некоторых ограничений), следовательно, часть размером  $b$  байт состоит из  $l = \frac{8 \cdot b}{w}$  слов. Всю резервную копию можно представить в виде матрицы  $D$ , где элемент  $d_{ij}$  –  $j$  слово  $i$  части резервной копии, а генерируемые части в виде матрицы  $C$ , где  $c_{ij}$  –  $j$  слово  $i$  генерируемой части.

Матрицу  $C$  получим из уравнения:

$$D \cdot F = C, \quad (1)$$

где  $F$  – матрица Вандермонда размерности  $m \times n$ , каждый элемент  $f_{i,j}$  которой вычисляется по формуле  $f_{i,j} = j^{i-1}$ . Поскольку вся левая часть уравнения (1) нам известна, мы сможем найти матрицу  $C$  (2).

Предположим, что ровно  $m$  частей было утеряно. Для начала вычислим матрицу  $A$  на основе матрицы Вандермонда размерности  $m \times n$ . Выполняя элементарные преобразования: перестановка местами любых двух столбцов матрицы, умножение любого столбца матрицы на константу, отличную от нуля, сложение столбцов, приведем матрицу Вандермонда к такому виду, что верхние  $n \times n$  элементов будут являться единичной матрицей. Определим матрицу  $E = \begin{bmatrix} D \\ C \end{bmatrix}$  и получим следующее уравнение (3).

$$\begin{bmatrix} f_{1,1} & f_{1,2} & \dots & f_{1,n} \\ f_{2,1} & f_{2,2} & \dots & f_{2,n} \\ \vdots & \vdots & & \vdots \\ f_{m,1} & f_{m,2} & \dots & f_{m,n} \end{bmatrix} \cdot \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,l} \\ d_{2,1} & d_{2,2} & \dots & d_{2,l} \\ \vdots & \vdots & & \vdots \\ d_{n,1} & d_{n,2} & \dots & d_{n,l} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,l} \\ c_{2,1} & c_{2,2} & \dots & c_{2,l} \\ \vdots & \vdots & & \vdots \\ c_{m,1} & c_{m,2} & \dots & c_{m,l} \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \dots & n^{m-1} \end{bmatrix} \cdot \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,l} \\ d_{2,1} & d_{2,2} & \dots & d_{2,l} \\ \vdots & \vdots & & \vdots \\ d_{n,1} & d_{n,2} & \dots & d_{n,l} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,l} \\ c_{2,1} & c_{2,2} & \dots & c_{2,l} \\ \vdots & \vdots & & \vdots \\ c_{m,1} & c_{m,2} & \dots & c_{m,l} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \dots & n^{m-1} \end{bmatrix} \cdot \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,l} \\ d_{2,1} & d_{2,2} & \dots & d_{2,l} \\ \vdots & \vdots & & \vdots \\ d_{n,1} & d_{n,2} & \dots & d_{n,l} \end{bmatrix} = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,l} \\ d_{2,1} & d_{2,2} & \dots & d_{2,l} \\ \vdots & \vdots & & \vdots \\ d_{n,1} & d_{n,2} & \dots & d_{n,l} \\ c_{1,1} & c_{1,2} & \dots & c_{1,l} \\ c_{2,1} & c_{2,2} & \dots & c_{2,l} \\ \vdots & \vdots & & \vdots \\ c_{m,1} & c_{m,2} & \dots & c_{m,l} \end{bmatrix} \quad (3)$$

Вычеркивая из матриц А и Е частей, соответствующих утерянным частям, получаем матрицы А' и Е'. Получаем новое уравнение:

$$A' \cdot D = E' \quad (4)$$

По определению, все строки матрицы Вандермонда F линейно независимы. Поскольку матрица А была получена из F путем элементарных преобразований, ее строки также линейно независимы, а матрица А' является обратимой, и, следовательно, уравнение (4) имеет решение. Решив (4), мы получим матрицу D, а, следовательно, восстановим резервную копию.

### Принцип работы программы

Запуск программы резервного копирования происходит в автоматическом режиме одновременно со

стартом операционной системы. При первой загрузке будет предложено отредактировать настройки по умолчанию. Рекомендуется, по крайней мере, выбрать папку с файлами, резервные копии которых необходимо создавать, и установить расписание работы.

Запущенная программа является одновременно и сервером и клиентом. Начав работу, она регистрирует PNRP имя и создает WCF службу, использующую IP-адрес и порт, указанный в настройках. В дальнейшем, любая другая копия программы, используя PNRP имя, сможет обнаружить эту службу, установить соединение и начать обмен данными.

Когда наступит время, заданное расписанием, программа упакует все папки и файлы в один ZIP файл. В конец получившегося файла, будут дописаны нули, чтобы его размер стал кратен одному мегабайту. Формат ZIP устроен так, что данная операция никак не скажется на его работоспособности. После

чего файл будет разбит на части, и с помощью алгоритма RS-RAID программа сгенерирует дополнительные части, которые создадут избыточность информации и увеличат тем самым надежность всей системы. Указав PNRP имя, программа получит список всех доступных WCF служб и информацию, необходимую для подключения к ним. Используя полученную информацию, она установит соединения и разошлет случайным образом все имеющиеся части. Возникающие исключения, общее время работы и информацию о результате записываются в файл журнала. Рассылка и прием частей от других компьютеров могут осуществляться одновременно, так как производятся в разных потоках.

Важно помнить, что задачей системы является резервное копирование, а не постоянная архивация данных, поэтому встает вопрос об удалении устаревших копий. Время, по истечении которого резервную копию можно считать устаревшей, задается в настройках, по умолчанию оно равно трем дням. После успешного завершения работы, программа посылает всем копиям команду удалить устаревшие части.

В случае утери данных, система предоставит пользователю список всех доступных для восстановления резервных копий, их размер и дату создания. При выборе одной из них, все части этой копии будут загружены, ZIP файл восстановлен и распакован.

### Список литературы

1. Кузнецов Л.К., Ноздрин Ю.И. Защита информации и информационная безопасность: Учеб. пособие. – М.: ВГНА, 2005. – 236 с.
2. .NET Сетевое программирование для профессионалов /Паркер Т., Кровчик Э., Кумар В., Лагари Н. – М.: Лори, 2005. – 400с.
3. Using P2P in backup and storage systems – [http://deptinfo.unice.fr/~dalle/wiki/uploads/Enseignements/Cours\\_Ubi\\_Part1.pdf](http://deptinfo.unice.fr/~dalle/wiki/uploads/Enseignements/Cours_Ubi_Part1.pdf)
4. Plank J. A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems – <http://web.eecs.utk.edu/~plank/plank/papers/CS-96-332.pdf>