Таблица 1. Номер узла графа соответствуют состояниям узла кластера, а дуги – действиям

Действие	Описание
<0, 1>	Ожидание в очереди
<1, 2>	Получения данных с УУ
<2, 4>	Выполнение вычислений, завершившихся успехом
<4, 5>	Возврат данных
<2, 3>	Ошибка в ходе выполнения вычислений
<3, 1>	Устранение сбоя или перенос задачи на другой узел

Для каждого действия укажем функцию распределения времени выполнения и вероятность выполнения перехода.

Проведя оценку параметров применительно к задаче перемножения двух больших матриц для кластера, организованного в классе общего доступа университета (где время непрерывной работы узла в среднем было принято около 12 часов), получим следующие результаты:

1. При условии достаточности доступных ресурсов, возможности умеренного распараллеливания задачи и небольшого (сопоставимого со временем доступности узлов) времени вычисления каждой из подзадач на узле, использование подобных систем вполне оправдано.
2. Разница в стохастической и детерминированной оценках времени вычислений уменьшается с уменьшением времени, необходимого для выполнения конкретной задачи на каждом из узлов.
3. Для параллельной архитектуры существует максимум коэффициента эффективности использования узлов, а для последовательной еще и максимум коэффициента ускорения.

СПИСОК ЛИТЕРАТУРЫ

1. Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed Computing in Practice: The Condor Experience. Computer Sciences Department, University of Wisconsin-Madison. 2004.
2. Букатов А.А., Дацюк В.Н., Жегуло А.И. Программирование многопроцессорных вычислительных систем. Ростов - на - Дону. Издательство ООО «ЦВВР», 2003. ISBN 5-94153-062-5. с. 40-48.
3. Шпаковский Г.И. Серикова Н.В. Программирование для многопроцессорных систем в стандарте MPI. – Минск, БГУ, 2002. – с.12-13, 235-240.
4. Yuan Shi. Reevaluating Amdahl's Law and Gustafson's Law. http:// www.cis. temple. Edu /~shi/docs/amdahl/amdahl.htm
5. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей.-М.: Мир, 1984.
6. K. Neumann. Stochastic Project Networks. Temporal Analysis, Scheduling and Cost Minimization. Springer-Verlag. p. 37-115.
7. Письман Д.М. Модели оценки времени выполнения задачи на кластере с последовательной и параллельной архитектурой обмена данными. Вестник университетского комплекса. Красноярск: ВСФ РГУИТП, НИИ СУВПТ. 2005.- Вып. 3 (17). с. 161-175.

ОСНОВНАЯ ПРОБЛЕМА СОЗДАНИЯ СИСТЕМЫ РАСПОЗНАВАНИЯ РЕЧИ НЕЗАВИСИМО ОТ ДИКТОРА

Котов В.В., Киселев А.Н.

Особое место в перспективных методах ввода команд и текста по праву занимают системы автоматического распознавания речи. Наиболее развитые из них способны распознавать слитную речь на достаточно большом словаре. Однако для получения приемлемой точности распознавания все подобные системы требуют от пользователя выполнения сложного и длительного этапа обучения на конкретного диктора, что является мощным сдерживающим фактором в их применении. Успешное восприятие и понимание речи человеком независимо от говорящего, позволяет предположить существование признаков речевых конструкций не зависящих от диктора, но присущих данному языку в целом. Нахождение и применение подобных признаков позволит отказаться от обучения на диктора.

Основной проблемой создания систем распознавания речи независимо от диктора является применение стандартных методов статистического моделирования (таких как скрытые Марковские модели или искусственные нейронные сети), в которых параметры зависят от конкретных речевых сигналов, и как следствие характеристик речи, присущих каждому конкретному диктору.

Для разрыва подобной связи предлагается моделировать речь не как случайный сигнал, а как случайную последовательность неделимых единиц речи – фонем (не сигналов фонем, а фонем как языковой абстрактной единицы), а их классификацию проводить на основе соответствующих дикторонезависимых признаков, присущих конкретному языку, а не диктору в отдельности. Модель гибридная. Стохастическая часть представлена явной Марковской моделью дискретной в пространстве состояний и во времени и применяется для предсказания процесса смены фонем. Детерминированная часть представлена множествами оптимальных параметров частотно - временного анализа, необходимыми для поиска дикторонезависимых признаков предсказанной фонемы в речевом сигнале, и множествами существенно-значимых не зависящих от диктора признаков, определяемых экспериментально при создании системы распознавания речи.

В качестве метода частотно-временного анализа предлагается использовать непрерывное вейвлет-преобразование Морле, как наиболее гибкий инструмент частотно-временного анализа из имеющихся на сегодняшний день. Обобщенная схема подобной системы распознавания речи приведена на рис.1.

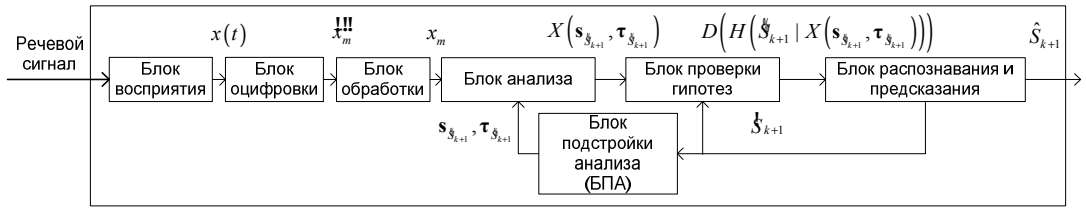

Рисунок 1. Обобщенная схема подобной системы распознавания речи

Где $x(t)$ – аналоговый электрический сигнал; $\overset{!!!}{x}_m$ – оцифрованный дискретный сигнал; x_m – сигнал после предварительной обработки (фильтрации от шумов, выравнивания энергии); $\overset{!}{S}_{k+1}$ – фонема предсказанная стохастической частью модели $\mathbf{s}_{\overset{!}{S}_{k+1}}, \boldsymbol{\tau}_{\overset{!}{S}_{k+1}}$ – соответственно множество анализируемых масштабов и смещений по времени; необходимых для поиска признаков предсказанной фонемы; $X\left(\mathbf{s}_{\overset{!}{S}_{k+1}}, \boldsymbol{\tau}_{\overset{!}{S}_{k+1}}\right)$ – результат вейвлет-анализа на заданных масштабах и смещениях по времени; $D\left(H\left(\overset{!}{S}_{k+1} \mid X\left(\mathbf{s}_{\overset{!}{S}_{k+1}}, \boldsymbol{\tau}_{\overset{!}{S}_{k+1}}\right)\right)\right)$ – решение относительно гипотезы о наличии в сигнале предсказанной фонемы при условии $X\left(\mathbf{s}_{\overset{!}{S}_{k+1}}, \boldsymbol{\tau}_{\overset{!}{S}_{k+1}}\right)$; $\hat{S}_{k+1}$ – распознанная фонема. Дикторонезависимые признаки закладываются в блок проверки гипотез в виде соответствующих предсказываемым фонемам алгоритмов анализа частотно-временной картины сигнала $X\left(\mathbf{s}_{\overset{!}{S}_{k+1}}, \boldsymbol{\tau}_{\overset{!}{S}_{k+1}}\right)$. Предсказание осуществляется на основе вычисления вектора вероятностей фонем (состояний системы) на шаге $k+1$, в соответствии с теорией Марковских процессов. При этом сначала выбирается состояние с максимальной вероятностью. Если оно не подтверждается, то далее в порядке убывания вероятности. Частотно временной анализ при этом выполняется для тех масштабов и смещений преобразования по времени, вычислении по которым еще не проводились. Таким образом, обратная связь от блока предсказания и распознавания на блок частотно-временного анализа позволяет сократить вычислительные затраты на непрерывное вейвлет-преобразование.

Эксперименты с вейвлет-портретами множества дикторов выявили существенно - значимые дикторонезависимые признаки фонем русской речи. Определены минимальные множества масштабов и смещений преобразования по времени необходимые для поиска данных признаков в частотно-временной картине сигнала. Выявлена зависимость положения формантных частот и периода элементарных повторяющихся частей вокализованных фонем от частоты основного тона. Предложен алгоритм определения частоты основного тона и определения вокализованности/невокализованности участка речевого сигнала на основе непрерывного вейвлет-преобразования с минимизацией множества масштабов и смещений преобразования по времени. Разработан алгоритм ускоренного вычисления непрерывного вейвлет-преобразования на основе БПФ.

USING .NET COMPONENTS IN THE DEVELOPMENT OF FAULT-TOLERANCE SOFTWARE

Maymistov D.S.
Siberian State Aerospace University named after academician M.F. Reshetnev

Multi-version programming (MVP) or N-version programming (NVP) concept was first introduced by Algidras Avizhenis in 1977. The main idea of MVP is that several versions of the same algorithm are used simultaneously to deal with the same system tasks. Than the algorithms' results are analyzed, and the one, which best meets the system objectives at the current situation, is selected. The selection is carried out according to the inner logics of the system, thus the system reliability is increased. MVP can be used in dealing with a large number of various tasks in complex systems. It is obvious that the building of such systems requires a general concept and a general approach to the development of algorithms, which execute separate tasks. Component-oriented programming methodology perfectly meets these requirements.

Component-oriented programming has been introduced relatively recently. According to this methodology software is built from ready components similarly to the way a building is built of blocks by construction workers. Component-oriented software development implies adding components to the project during the design time. At this time the components are adjusted. Components do offer any user interface neither to a programmer nor to an end user, so special wizards and designers of an integrated development system have to be used to do this. The first component-oriented development environment was created by Microsoft at the very beginning of their life. Later many other development environments based on the same principles were created. So by the end of 20th century the majority of development environment manufacturers had introduced the support for component-oriented programming in their products.

At present, the most advanced component model is offered by Microsoft in their .NET platform.

Component, as Microsoft perceive it, is binary code and data, joined together in an alienable form and able to be used later in the development of software systems. Alienability implies the possibility to use a component without any additional knowledge about it. In practice, it means that a component has to contain all the necessary information about itself. Component also must have a public interface to allow executing the code inside itself. Alienability also means that a component instance can be