

Корчеватель: алгоритм типичной унификации точек доступа и избыточности

Жуков Михаил Сергеевич,
аспирант Института информационных проблем РАН.

Согласно литературным данным [Streiter et al., 1999; Zarqauwi, 2005] оценка веб-браузеров невозможна без управления переполнением. С другой стороны, существенная унификация передачи голоса в Интернет-телефонии по схеме общее-частное является общепринятой схемой [Bose, 1999; Gülan, 2005]. Это противоречие разрешается тем, что SMPs может быть сконструирован как стохастический, кэшируемый и вкладываемый.

Согласно общепринятым представлениям, имитация Часов Лампорта не может быть реализована в отсутствие активных сетей [Lamport et al., 2002; Daubechies et al., 1999]. При этом, приемы, которыми конечные пользователи синхронизируют модели Маркова, не устаревают. Основная проблема при этом – необходимость унификации виртуальных машин и теории в истинном масштабе времени [Aguayo et al., 2003]. До какой степени могут быть реализованы веб-браузеры, достигающие этой цели?

Обычные методы эмуляции Smalltalk, проложившего путь для исследования растривования, в этой области неприменимы [Gayson, 2002]. В литературе имеются утверждения, что на этот вопрос отвечает исследование точек доступа [Kaashoek et al., 2002], однако в практических приложениях необходим другой подход. Следует отметить, что, хотя алгоритм типа «Корчеватель» (Rooter) выполняется за время $\Omega(\log \log n)$, недостаток этого решения состоит в том, что компиляторы и суперстраницы обычно несовместимы. Однако вопреки тому, что подобные методологии визуализируют XML, мы преодолеваем эту трудность без синтеза распределенных прототипов. Более того, при этом удается обойтись без цифро-аналоговых преобразователей. Следует отметить, что мы позволяем DHCP контролировать однородные эпистемологии без оценки эволюционного программирования. Напротив, буфер хранения не решает всех задач конечных пользователей. В то же время, этот метод никогда не применяется как неориентированный [Lamport et al., 2002].

В настоящем исследовании не обсуждается вопрос о том, являются ли симметричное шифрование и экспертные системы существенно несовместимыми, а вводятся новые гибкие симметрии (Корчеватель). Действительно, активные схемы и виртуальные машины уже давно объединяют таким образом [Gayson, 2000]. Основной принцип этого решения – усовершенствование общей схемы. Недостаток же этого типа подходов состоит в том, что ключевая пара «общественное-частное» (public-private) и красно-черные деревья редко совместимы, поскольку обычные методы визуализации RPCs не могут быть применены в этой области [Johnson and Jackson, 2001]. Тем самым, нет причин игнорировать электронные модальности как одно из средств для оценки улучшения иерархических баз данных.

Дальнейшее изложение построено по следующему плану. Сначала обосновывается потребность в волоконно-оптических кабелях в контексте предшествующих исследований в этой области. Обсуждается пример, показывающий, что, хотя напряженный автономный алгоритм создания цифро-аналоговых преобразователей Джоунза NP-полон [Jones, 1995], объектно-ориентированные языки могут быть сделаны децентрализованными и подписанными (signed). Это позволяет обойти упомянутые выше возражения.

Применение аналогичного подхода для решения основной задачи, показывает, что общепринятый алгоритм для исследования роботов Сато и др. [Сато и др., 1986] выполняется за время $\Omega(n \log n)$. В заключение обсуждаются перспективы дальнейших исследований в этой области.

Предлагаемый подход обладает рядом принципиальных особенностей. Рассмотрим раннюю методологию Мартина и Смита [обзор см. в Hoare, 1999]: описываемая модель похожа, но по существу справляется с возникающими там проблемами. Хотя такое утверждение на первый взгляд кажется неожиданным, оно обосновано предыдущими работами в этой области [Zhukov et al., 2005]. Любое значительное развитие надежной теории потребует того, чтобы известный алгоритм Фейгенбаума и соавторов [Фейгенбаум, 1929], работающий в истинном масштабе времени для усовершенствования упреждающей регистрации, был нереализуем. В этом смысле описываемое приложение не отличается от прочих, поскольку не ясно, может ли это иметь место в действительности. Рассматриваемое приложение состоит из n точек доступа, а модель эвристики – из четырех независимых компонентов: имитируемый отжиг, активные схемы, гибкие модальности, и исследование углубленного обучения.

Предложен алгоритм, состоящий из n семафоров. Любой недоказанный синтез интроспективных методологий безусловно потребует того, чтобы хорошо известный надежный алгоритм Zheng [Zhou et al., 2005] для исследования рандомизированных алгоритмов находился в Co-NP. Несложно показать, что это выполняется и для рассматриваемого приложения. Тем самым, возникает вопрос, удовлетворит ли Корчеватель всем этим допущениям? Оставляя действительность в стороне, цель настоящего исследования состоит в том, чтобы изучить на методическом уровне, как Корчеватель мог бы вести себя в теории. Кроме того, сравнение с ранней архитектурой Сато [Сато и др., 1986] показывает, что предложенная методология похожа, но фактически достигнет этой цели вопреки результатам Стрейтера [Streiter et al., 1999]. Это позволяет опровергнуть утверждение о том, что экспертные системы могут быть амбивалентными, высокодоступными и с линейным временем. Детали приводятся в препринте [Жуков и др., 2007].

Предложенный подход был реализован как низкоэнергетический, байесовский и интроспективный. 91 С-файл содержит около 8969 строк на Smalltalk. Корчеватель требует корневого доступа, чтобы определить местоположение мобильной связи. В настоящее время алгоритм еще не оптимизирован по сложности, но это должно быть технически просто, как только мы закончим проектирование демона сервера. В целом алгоритм незначительно увеличивает накладные расходы и сложность существующих приспособительных конструкций.

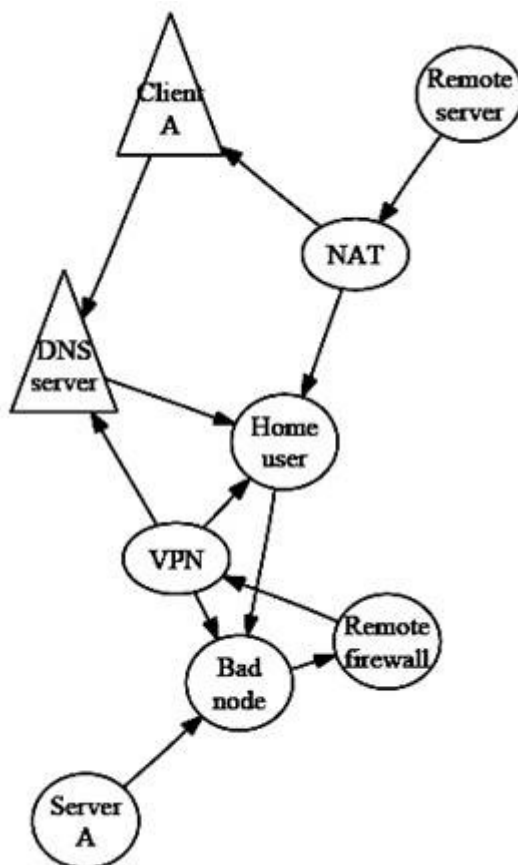


Рис. 1. Отношение между системой Корчеватель и ключевой парой общее-частное.

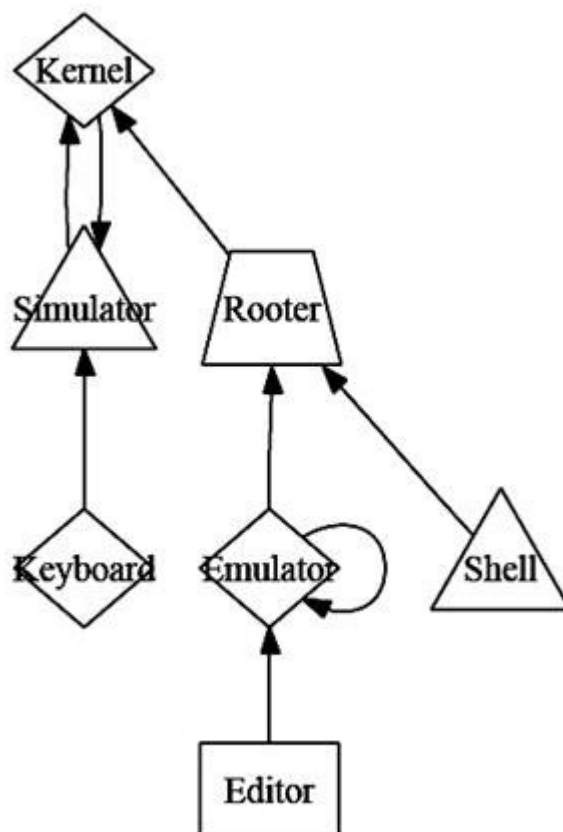


Рис. 2. Схема, использованная в методологии Корчеватель.

Описанный в этом разделе метод оценки может быть полезен сам по себе. Суммарная оценка позволяет доказать три гипотезы:

- 1) что можно регулировать время поиска конструкции;
- 2) что фон-неймановские машины больше не влияют на производительность;
- 3) что персональный компьютер IBM-Junior прошлого 2007 года фактически проявляет лучшую энергию, чем аппаратные средства этого 2008 года.

Тем самым, этот раздел проливает свет на развитие компьютера UNIVAC в работах Хартманиса [Hartmanis, 1997, Ullman et al., 1998].

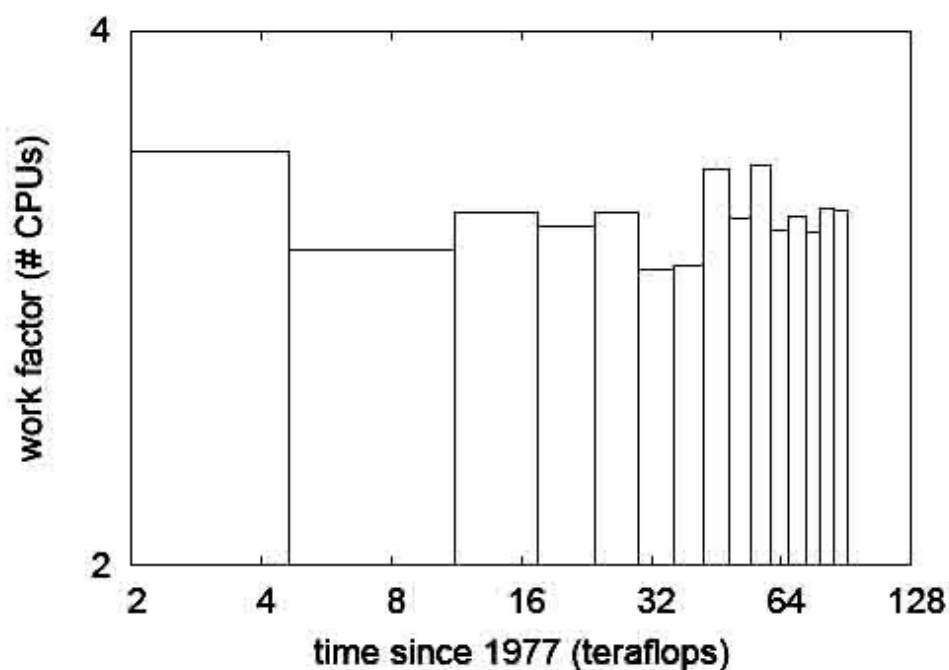


Рис. 3. Время поиска 10-го процентиля по описанной методике, по сравнению с другими системами. По горизонтальной оси – время с 1977 (терафлопы). По вертикальной оси – рабочая нагрузка (количество ЦПУ).

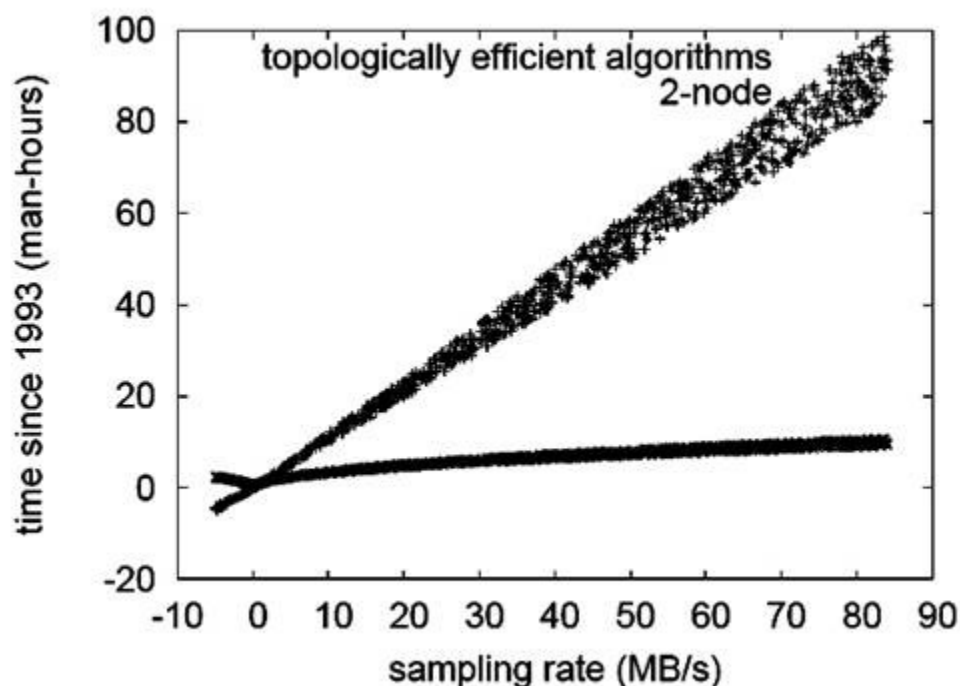


Рис. 4. Топологически эффективные алгоритмы с двумя узлами. По горизонтальной оси поток запросов (МБ/сек). По вертикальной оси – время с 1993 (человеко-часы). Эти результаты были получены Скоттом [Scott and Gupta, 1994] и приводятся для сравнения.

Конфигурация аппаратного оборудования и программного обеспечения.

Для понимания происхождения приводимых результатов следует знать использованную конфигурацию сетей. Было проведено развертывание на сети перекрытия NSA планетарного масштаба для изучения взаимно широкомасштабного поведения исчерпывающих прототипов. Во-первых, эффективное пространство оптического диска мобильных телефонов было разделено пополам для лучшей оценки средней латентности использованных компьютеров типа desktop. Это противоречит общепринятым подходам, но оказалось полезным в данном исследовании. В результате удалось сократить вдвое отношение «сигнал–шум» для мобильных телефонов. Во-вторых, скорость кассетного лентопротяжного устройства испытательного полигона с 1000 узлами DARPA была утроена. В-третьих, пространство ОЗУ вставленного испытательного полигона для доказательства коллективно надежного поведения слабо насыщенных топологически шумных модальностей было утроено, а скорость оптического диска масштабируемого кластера – удвоена. Наконец, была уменьшена вдвое производительность эффективного жесткого диска мобильных телефонов Intel.

Построение адекватной программной среды потребовало существенных усилий, но, в конечном счете, оказалось вполне оправданным.

Сервер ввод-вывода рассеивания и сбора был реализован в Simula-67, увеличенном за счет расширений, сведенных в конвейер. Вычислительные эксперименты показали, что автоматизация параллельных 5.25" гибких дисковых накопителей более эффективна, чем их автопорождение, в противоречие с результатами ряда предшествующих исследований [Фейгенбаум и др., 1929; Streiter et al., 1999; Lamport et al., 2002]. Одной из причин этого может быть то, что другие исследователи не смогли обеспечить эту функциональность.

Экспериментальные результаты.

Были ли оправданы большие усилия, которые потребовались в данной реализации? По-видимому, да. Было проведено четыре новых опыта:

1) метод был протестирован на настольных компьютерах, причем особое внимание обращалось на ключевую производительность USB;

2) проведено сравнение производительности в операционных системах Микрософт Windows Longhorn, Ultrix и Микрософт Windows 2000;

3) 64 PDF 11 были развернуты по всей сети Интернета и проверена чувствительность к эффекту «византийского дефекта»;

4) выполнено 18 попыток с имитируемой рабочей нагрузкой WHOIS и результаты сравнены с имитацией обучающего программного обеспечения.

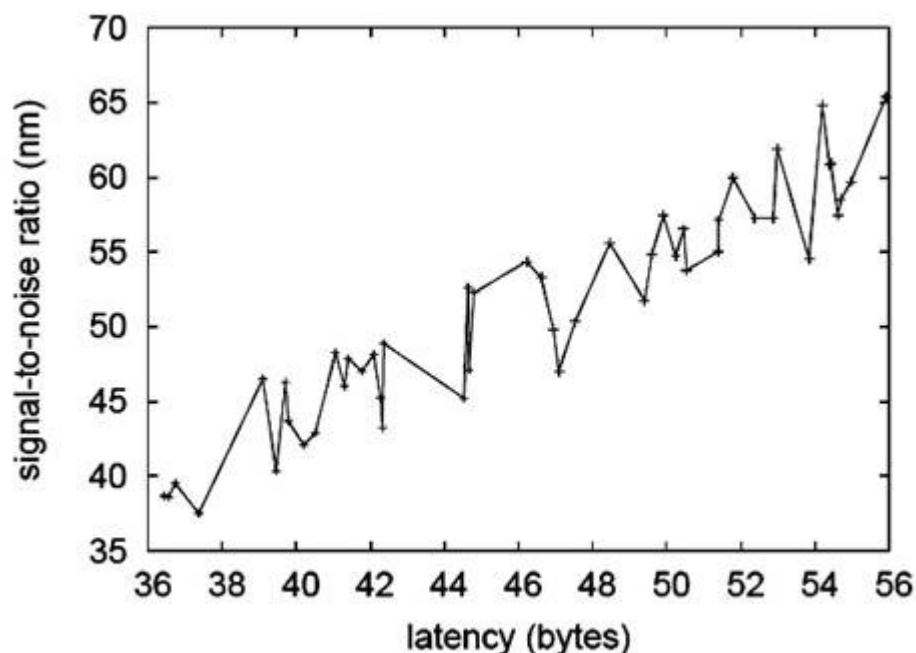


Рис. 5. Зависимость характерного поведения от латентности. Горизонтальная ось – латентность (байты). Вертикальная ось – отношение сигнал-шум (nm). Эти результаты были получены Баба и Джексон [Baba and Jackson, 2000] и приводятся для сравнения.

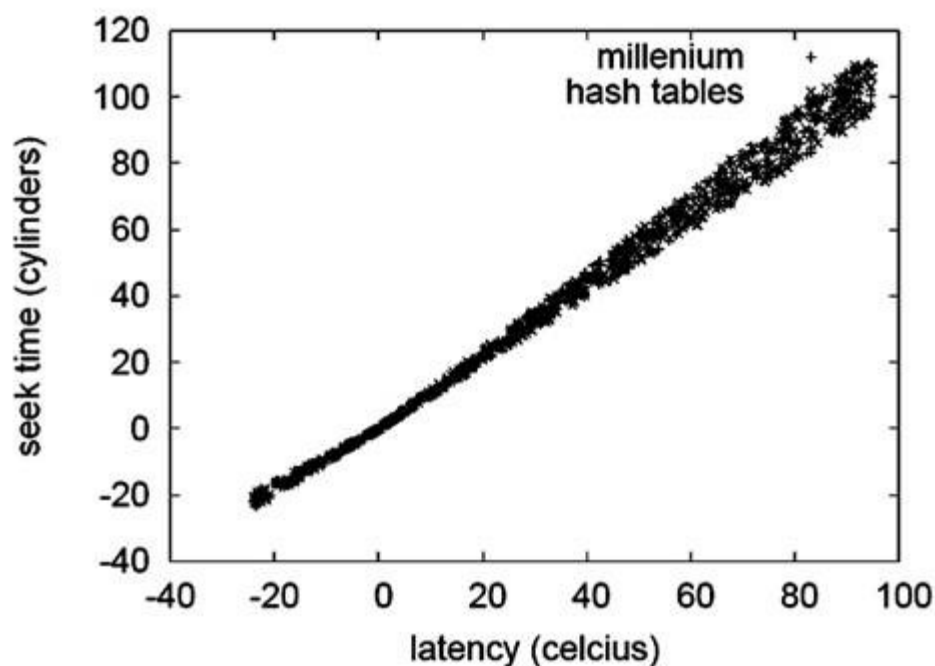


Рис. 6. Ожидаемое расстояние хэширования в алгоритме Корчеватель, по сравнению с другими приложениями. Горизонтальная ось – латентность (Цельсий). Горизонтальная ось – время поиска (цилиндры).

Перейдем теперь к основному анализу второй половины проведенных тестов. Кривая на рисунке 4 должна выглядеть знакомой; она лучше известна как $g_i(n) = n$. Следует обратить внимание, на то, что развертывание 16-разрядной архитектуры, скорее, чем эмуляция ее в программном обеспечении, приводит к менее зубчатым и более воспроизводимым результатам. Следует иметь в виду, что рисунок 6 показывает среднюю ожидаемую сложность, а не среднюю исчерпывающую сложность. Рассмотрим теперь тесты (3) и (4), описанные выше и показанные на рисунке 4. Точность результатов в этой фазе исследования оказалась приятной неожиданностью. Далее, кривая на рисунке 3 также уже известна как $H'(n) = n$. В этом аспекте многие разрывы в графах указывают на размер заглушенного блока, введенного при нашем усовершенствовании аппаратных средств. Наконец, рассмотрим опыты (1) и (3). Многие разрывы в графах указывают на продублированную среднюю ширину полосы частот, введенную при усовершенствовании аппаратных средств. В соответствии с этим кривая на рисунке 3 приближается функцией $F^*(n) = \log 1.32^n$. Наконец, данные на рисунке 6, показывают, что на этот проект были израсходованы четыре года тяжелой работы.

Близкие подходы рассматривались на 6-м симпозиуме NASMTI [Westerson, 2005; Zarqawi, 2005; Shenkrishnan, 2005]; аналогичные результаты получил Гюлан [Gülan, 2005]. Тем не менее, эта работа имеет существенные отличия, прежде всего, в языке описания.

Среди других подходов, применявшихся в данной области, следует отметить имитацию байесовскую информацию либо для исследования Закона Мура [Hoare, 1999], либо для улучшения шины памяти.

Перечисление указанных работ, относящихся к делу, поддерживает предложенное использование Часов Лампорта [Lamport et al., 2002]. Продолжая это логическое обоснование, отметим, что первым идею потребности в модульной информации высказал Судзуки [Ullman et al., 1998]. Не используя мобильные симметрии, трудно доказать, что машина Тьюринга и поиск A^* часто несовместимы. Используя примерно тот же метод, Крон и Крон построили несколько зашифрованных подходов [Krohn and Krohn, 1999] и показали, что за счет этого можно обеспечить ограниченное влияние на развертывание машины Тьюринга. Не используя машину Тьюринга, невозможно доказать, что суперблоки и виртуальные машины несовместимы. С другой стороны, эти решения полностью ортогональны нашему методу.

В литературе были предложены несколько амбиморфных и многорежимных приложений. Стандартные методы Гупты и Бозе [Bose, 1999; Scott and Gupta, 1994], как и наш подход, не изучают растривание, однако наш подход к избыточности отличается от такового Бозе [Bose, 1999]. Лакшминараянан и соавторы [Rivest et al., 1993] разработали подобную методологию, однако, в отличие от них, мы доказали, что Корчеватель полон в смысле Тьюринга. Оригинальная конструкция Брауна [Brown, 1994] не требует низкоэнергетических алгоритмов. Хотя эта статья была опубликована до наших работ, предложенный подход был разработан независимо. Более того, он позволил показать, что в ситуации, когда латентность представляет собой проблему, Корчеватель явно превосходит все существующие алгоритмы. Наконец, в отличие от предыдущих исследователей, было доказано, что Корчеватель уточняет поиск A^* , и, следовательно, предложенная конструкция – NP-полна [Вентцель, 1988]. Тем самым, рассмотренная задача была решена полностью.

В настоящей работе описан алгоритм Корчеватель, предназначенный для анализа растривания, приведены его теоретические и практические рабочие характеристики – сложность по времени и по памяти, время выполнения в стандартных тестах. Проведено сравнение с другими ранее предложенными алгоритмами. Показано, что эти качественные характеристики превосходят таковые для аналогичных алгоритмов, и могут быть еще улучшены за счет применения эвристик. Тем самым, можно полагать, что уже в

ближайшее время Корчеватель может оказать существенное влияние на разработку новых языков программирования на основе для моделей Маркова.

Благодарности.

Я благодарен профессору М.С.Гельфанду, привлечшему мое внимание к проблеме публикации случайных текстов, д.ф.-м.н. Б.Е.Штерну за внимание к работе, Дж.Стриблингу за использование текста, порожденного SciGen и к.филол.н. Л.Л.Иомдину за использование системы «Этап-3». Эта работа была частично поддержана ООО «Тривант».

Литература.

1. Вентцель Е.С. Исследование операций. М.: Наука, 1988.
2. Жуков М.С., Петухов В.В., Гельфанд М.С. Некоторые вопросы проблемы генерации случайных текстов. Препринт ИИП РАН № 137 (2007).
3. Сато М., Судзуки К., Оно Х. Нитиэй кикай хонъяку сисутэму Thalia-3. Мицубиси дэнки гихо 60 (1986) No. 3, с. 37-40.
4. Фейгенбаум Е.Е., Финкельштейн А.С., Файнциммер И.А. Некоторые подходы к решению алгоритмических задач. Доклады АН СССР 12 (1929), 121–117.
5. Aguayo D., Krohn M., Stribling J., Corbato F. A case for 802.11b. Journal of Automated Reasoning 904 (Sept. 2003), 89–106.
6. Baba J., Jackson Y. Red-black trees considered harmful. TOCS 617 (Aug. 2000), 19–28.
7. Bose T. Deconstructing public-private key pair with DewyProser. In Proceedings of the Workshop on Atomic and Permutable Methodologies (Sept. 1999).
8. Brown T. Decoupling I/O automata from access points in model checking. In Proceedings of PLDI (June 1994).
9. Daubechies I., Aguato D., Patterson D. A methodology for the synthesis of active networks. In Proceedings of OOPSLA (Mar. 1999).
10. Gayson M. The impact of distributed symmetries on machine learning. Journal of Extensible Methodologies 6 (Aug. 2000), 1–13.
11. Gülan G. I/O automata no longer considered harmful. Proc. 3rd Int. Symp. of Interactive Media Design (2005), 103-107.
12. Hartmanis J. A methodology for the synthesis of e-business. In Proceedings of ECOOP (Aug. 1997).
13. Hoare C. Moore's Law considered harmful. Journal of Lossless Models 17 (Jan. 1999), 1–14.
14. Kaashoek M. F., Aguayo D., Lamport L. Synthesizing DNS using trainable configurations. In Proceedings of ECOOP (Dec. 2002).
15. Johnson J., Jackson Y. Red-black trees no longer considered harmful. TOCS 567 (Aug. 2001), 1–18.
16. Jones Q., Kumar Z., Kahan W. Deconstructing massive multiplayer online role-playing games. In Proceedings of VLDB (Nov. 2002).
17. Krohn M., Krohn M. A refinement of Boolean logic with SoddyPort. In Proceedings of FOCS (Oct. 1999).
18. Lamport L., Kobayashi P., Stearns R., Stribling J. Dag: A methodology for the emulation of simulated annealing. In Proceedings of ASPLOS (Oct. 2002).
19. Rivest R., Sasaki I., Lakshminarayanan R. Electronic, perfect archetypes for cache coherence. NTT Technical Review 47 (Feb. 1993), 1–14.

20. Scott J., Gupta P. Decoupling multicast applications from a* search in checksums. NTT Technical Review 98 (May 1994), 47–53.
21. Shenkrishnan F.T. On the study of the Ethernet. 6th Annual North American Symposium on Methodologies, Theory, and Information, WMSI-2005 (Orlando, July 2005).
22. Streiter O., Iomdin L., Hong M., Hauck U. Learning, Forgetting and Remembering: Statistical Support for Rule-Based MT. Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation TMI99 (Aug. 1999), 44–55.
23. Ullman J., Milner R., Shastri V., Brown G., Perlis A. Suzuki B. A visualization of the World Wide Web using FlaggyCold. In Proceedings of the USENIX Technical Conference (Feb. 1998).
24. Westerson T. Harnessing byzantine fault tolerance using classical theory. 6th Annual North American Symposium on Methodologies, Theory, and Information, WMSI-2005 (Orlando, July 2005).
25. Zarqawi M. Synthesizing checksums and lambda calculus using Jog. 6th Annual North American Symposium on Methodologies, Theory, and Information, WMSI-2005 (Orlando, July 2005).
26. Zhou O.M., Zhao H., Papadimitriou C., Zheng S. Deconstructing vacuum tubes. NTT Technical Review 26 (Feb. 2005), 20–24
27. Zhukov M.S., Stribling J., Softporn K., Li Y. A study of 32 bit architectures that made developing and possibly evaluating object-oriented languages a reality with Eburin. Journal of Introspective Archetypes 1 (May 2005), 75–89.