

© 2012 г. С.Н. АНДРИАНОВ, д-р физ.-мат. наук
(Санкт-Петербургский государственный университет, Санкт-Петербург)

ВИРТУАЛЬНЫЙ УСКОРИТЕЛЬ – РАСПРЕДЕЛЕННАЯ СРЕДА МОДЕЛИРОВАНИЯ И УПРАВЛЕНИЯ УСКОРИТЕЛЬНЫМИ КОМПЛЕКСАМИ

В данной работе рассматриваются теоретические и прикладные вопросы разработки распределенной среды, предназначенной как для моделирования, так и для управления ускорительными комплексами различного назначения. В основу работы положены строгие математические модели, а также современные методы и инструменты разработки распределенных информационных систем. В основу предлагаемых методов и технологий положен принцип максимальной взаимной адекватности математического аппарата и соответствующих программных средств и технологий.

VIRTUAL ACCELERATOR – A DISTRIBUTE ENVIRONMENT FOR ACCELERATOR FACILITIES MODELING AND CONTROL / S.N. Andrianov (Saint Petersburg State University, University ave. 35, Petrodvoretz, Saint Petersburg, 198504, E-mail: sandrianov@yandex.ru) In this paper we discuss some theoretical and practical problems of distributed environment development, devoted to both modeling and control of the accelerator facilities. In the basis of this work we put the rigorous mathematical models, as well as modern methods and tools development of distributed information systems. The proposed methods and techniques based on the principle of maximum mutual adequacy of mathematical tools and related software and technology.

1. Введение

Современные системы управления сложными физическими установками, например, такими как ускорительные комплексы различного назначения, термоядерные реакторы и т. п., предполагают использование четко выраженных сценариев, в соответствии с которыми осуществляется поддержка рабочего режима соответствующей установки. Разработка рабочих сценариев, обеспечивающих работоспособность соответствующих установок, базируется как на формализованных математических моделях, описывающих соответствующие процессы, так и на эффективном программном обеспечении, реализующем соответствующие модели. Комплексная сложность самих установок приводит к многосоставным моделям, при формализации которых необходимо использовать широкий спектр математических формализаций и методов. Именно разнообразие моделей, их вычислительная сложность приводит к необходимости использования распределенных вычислительных средств и соответствующих технологий,

в частности Grid- и Cloud-технологий. При этом следует различать две компоненты подобного инструментария: теоретическую (физико-математическую) и программную. Первая компонента предназначена для предварительного исследования (теоретического и экспериментального) различных эффектов в рассматриваемой установке, в результате которого формируются сценарии, в рамках которых реализуется управление реальными установками. Вторая компонента отвечает за реализацию разработанного сценария как с целью проведения эффективных вычислительных экспериментов, так и обеспечения полученных режимов для управления реальными установками. Следует заметить, что как ускорительные комплексы, так и системы управления термоядерными установками в принципе не могут управляться в режиме реального времени в обычном смысле этого термина. Дело в том, что скорость протекания соответствующих процессов на несколько порядков выше возможностей реагирования соответствующих систем управления. Однако при настройке системы управления на рабочий режим осуществляется тестирование работоспособности системы управления с учетом данных, «снимаемых» с соответствующих контроллеров. Именно в этом смысле в научной литературе используется термин «виртуальный ускоритель» [1] и соответствующие реализации, например, с использованием хорошо известной системы управления сложными процессами и объектами – EPICS (Experimental Physical and Industrial Control System) [2]. В данной работе мы предлагаем расширение области применимости концепции «Виртуальный ускоритель» на обе компоненты, упомянутые выше.

В статье описаны разработанные математические модели, решения основных задач физики пучков, базирующиеся на различных моделях описания пучка частиц в терминах матричного формализма, предложенного автором данной статьи. Специфика решаемых задач приводит к необходимости расширения возможностей данного формализма, прежде всего, с целью повышения вычислительной эффективности. В работе обсуждаются методы и технологии, которые позволяют адаптировать предлагаемый математический аппарат для решения широкого класса задач физики пучков на основе его унификации в рамках концепции LEGO-объектов [3]. Подобные объекты являются минимальными математическими и компьютерными единицами, из которых «собирается» вычислительная модель конкретной задачи. При этом подобная сборка подчиняется единым правилам и может осуществляться как в символьной моде, так и в численной. Использование соответствующих баз данных и знаний позволяет не только повысить эффективность проведения вычислительных экспериментов, но и «накапливать» соответствующие данные и знания. С точки зрения организации вычислительных процедур подобная организация полностью «вкладывается» в концепцию SOA (Service Oriented Architecture). Таким образом, принцип модульности пронизывает все этапы процесса моделирования: от разработки математических модулей до соответствующих программных реализаций. Использование методов и технологий символьных вычислений позволяет не только накапливать формализованные знания в соответствующих базах данных и знаний, но и использовать их при выполнении вычислений с использованием параметрического представления соответствующей информации. Это, частности, позволяет использовать методы и технологии глобальной оптимизации, эффективность которых существенно зависит от форм представления функциональных критериев качества, ограничений и других, необходимых для решения подобных задач объектов.

2. Основные положения матричного формализма алгебраических методов Ли

Специфика широкого класса задач моделирования динамики пучков частиц в ускорительном канале позволяет использовать различные варианты теории возмущения. Одним из методов подобного типа является так называемый формализм алгебраических методов Ли, впервые введенный в теорию динамики пучков заряженных частиц в работах Алекса Драгта (см., например, [4]). В его основе лежит аппарат хронологических рядов Вольтерры [5]. Известно, что в стационарном случае выражение фазового потока через определяющее его векторное поле осуществляется с помощью экспоненциального отображения. Однако в нестационарном случае из-за не коммутирования векторных полей в разные моменты «времени» возникают достаточно серьезные трудности, что приводит к необходимости учитывать хронологический порядок соответствующих объектов в используемых разложениях. Для проведения необходимых оценок сходимости и построения конструктивных алгоритмов в работе используется метод Магнуса [8]. Вопросы сходимости и соответствующие оценки приведены во многих работах (см., например, [7]).

В основе динамики частиц лежат уравнения Ньютона–Лоренца, записанные в криволинейных системах координат (ассоциированной с так называемой опорной траекторией), как наиболее адекватных задачам физики пучков.

$$(1) \quad \frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}, t) = \sum_{k=1}^{\infty} \mathbb{P}_k(t) \mathbf{X}^{[k]}, \quad \mathbf{X}_0 = \mathbf{X}(t_0),$$

где $\mathbf{F} = \{F_1, \dots, F_{2n}\}^T$ и $\mathbf{X}^{[k]}$ вектор всех фазовых моментов k -го порядка для вектора фазовых переменных \mathbf{X} ($\dim \mathbf{X} = 2n$), например, $\mathbf{X} = \{x, p_x, y, p_y\}^T$ и \mathbb{P}^{1k} (with matrix size $\dim \mathbb{P}^{1k} = \binom{2n+k-1}{k}$) суть матрицы содержащие частные производные от вектор-функции \mathbf{F}

$$\{\mathbb{P}^{1k}(t)\}_{ij} = \frac{1}{k_1! \dots k_{2n}!} \frac{\partial^k F_i(x_j, t)}{\partial x_1^{k_1} \dots \partial x_{2n}^{k_{2n}}} \Big|_{x_1=\dots=x_{2n}=0}.$$

Уравнение (1) порождает операторное уравнение

$$(2) \quad \frac{d\mathcal{M}(t|t_0; \mathcal{L}_{\mathbf{F}})}{dt} = \mathcal{L}_{\mathbf{F}}(t) \circ \mathcal{M}(t|t_0; \mathcal{L}_{\mathbf{F}}), \quad \mathcal{M}(t_0|t_0; \mathcal{L}_{\mathbf{F}}) = \mathcal{I}d.$$

где $\mathcal{I}d$ – тождественный оператор, $\mathcal{L}_{\mathbf{F}}$ – Ли-оператор (оператор бесконечно малого сдвига вдоль опорной траектории) в соответствии с равенством [4]

$$\mathcal{L}_{\mathbf{F}} = \mathbf{F}^*(\mathbf{X}, t) \frac{\partial}{\partial \mathbf{X}} = \sum_{k=1}^{\infty} (\mathbf{X}^{[k]})^T \mathbb{P}_{1k}^T \frac{\partial}{\partial \mathbf{X}} = \sum_{k=1}^{\infty} \mathcal{L}_{\mathbf{F}_k}.$$

Преобразование Ли, порождаемое системами дифференциальных уравнений (1, 2), можно записать в виде

$$(3) \quad \mathcal{M}(t|t_0; \mathcal{L}_{\mathbf{F}}) = \text{T exp} \int_{t_0}^t \mathcal{L}_{\mathbf{F}}(\mathbf{X}, \tau) d\tau = \text{T exp} \sum_{k=1}^{\infty} \int_{t_0}^t \mathcal{L}_{\mathbf{F}_k} d\tau.$$

Здесь «Т» перед символом экспоненты обозначает так называемый хронологически упорядоченный экспоненциальный оператор Ли [7] (или оператор Дайсона) и k есть порядок соответствующего однородного полинома. Таким образом, решение уравнения (1) может быть найдено в два этапа. На первом этапе мы должны перейти к представлению Магнусса для хронологически упорядоченного оператора [7, 8]

$$(4) \quad \mathcal{M}(t|t_0; \mathcal{L}_{\mathbf{F}}) = \mathcal{M}\left(t|t_0; \sum_{t_0}^t \mathcal{L}_{\widehat{\mathbf{F}}_k}\right),$$

где $\widehat{\mathbf{F}}_k$ новые функции, которые могут быть вычислены в соответствии подходом Магнусса. На втором этапе мы должны воспользоваться формулой Зассенхауза [7] для представления нового экспоненциального оператора (4) в виде произведения экспоненциальных операторов более простой формы (например, ассоциированных с однородными полиномами $\mathbf{G}_k = \mathbb{G}_k \mathbf{X}^{[k]}$): $\prod_{k=1}^{\infty} \mathcal{M}\left(t|t_0; \sum_{t_0}^t \mathcal{L}_{\mathbf{G}_k}\right)$, $k \geq 1$. Реализуя соответствующие вычисления, мы можем получить аналитические соотношения для вычисления матричных коэффициентов соответствующих полиномов. Например, вводя обозначение [7] $\mathbb{P}_m^{k1} = \prod_{j=1}^k \mathbb{G}_m^{\oplus((j-1)(m-1)+1)}$ (здесь \oplus обозначает Кронекерову сумму), можно записать

$$(5) \quad \exp(\mathcal{L}_{\mathbf{G}_m}) \circ \mathbf{X} = \mathbf{X} + \sum_{k=1}^{\infty} \frac{\mathbb{P}_m^{k1}}{k!} \mathbf{X}^{[k(m-1)+1]}.$$

Проводя подобные вычисления мы получим выражения для представления «укороченного» оператора Ли. Например, для третьего порядка можно получить

$$(6) \quad \mathcal{M}_{\leq 3} \circ \mathbf{X} = \mathbb{R}^{11} \left(\mathbf{X} + \sum_{m=2}^3 \sum_{k=1}^{\infty} \frac{\mathbb{P}_m^{k1}}{k!} \mathbf{X}^{[k(m-1)+1]} + \sum_{l=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{k!l!} \mathbb{P}_2^{kl} \mathbb{P}_3^{l(k+1)} \mathbf{X}^{[2l+k+1]} \right),$$

Повторяя описанную процедуру, можно вычислить $\mathcal{M}_{\leq k} \circ \mathbf{X}$ для любого k . Однако, так как $\mathcal{M}_{\leq k}$ является укороченной формой \mathcal{M} , при вычислении которой мы использовали информацию о производящей функции только до k -го порядка включительно, то и результат действия $\mathcal{M}_{\leq k}$ на фазовый вектор \mathbf{X} мы должны вычислять только до членов k -го порядка включительно. Действуя аналогично получим

$$(7) \quad \mathcal{M}_{\leq 2} \circ \mathbf{X} \approx \mathbb{M}^{11} (\mathbf{X} + \mathbb{P}_2^{11} \mathbf{X}^{[2]}) = \mathbb{M}^{11} \mathbf{X} + \mathbb{M}^{12} \mathbf{X}^{[2]}, \quad \mathbb{M}^{12} = \mathbb{M}^{11} \mathbb{P}_2^{11},$$

$$(8) \quad \mathcal{M}_{\leq 3} \circ \mathbf{X} \approx \mathbb{M}^{11} \left(\mathbf{X} + \mathbb{P}_2^{11} \mathbf{X}^{[2]} + \left(\mathbb{P}_3^{11} + \frac{1}{2!} \mathbb{P}_2^{21} \right) \mathbf{X}^{[3]} \right) = \mathbb{M}^{11} \mathbf{X} + \mathbb{M}^{12} \mathbf{X}^{[2]} + \mathbb{M}^{13} \mathbf{X}^{[3]},$$

$$(9) \quad \mathbb{M}^{13} = \mathbb{M}^{11} \left(\mathbb{P}_3^{11} + \frac{1}{2!} \mathbb{P}_2^{21} \right)$$

и т. д. Итак, последовательно действуя операторами \mathcal{M}_k , $k < m$ на вектор \mathbf{X} и оставляя в получаемых выражениях члены вплоть до k -го порядка включительно, мы осуществим процедуру построения матричного представления оператора в базисе Пуанкаре–Витта \mathbf{X} , $\mathbf{X}^{[2]}$, $\mathbf{X}^{[3]}$, ..., $\mathbf{X}^{[k]}$, ... Иными словами, мы можем написать

$$(10) \quad \mathcal{M} \circ \mathbf{X} = \mathbb{M} \mathbf{X}^{\infty} = (\mathbb{M}^{11} \mathbb{M}^{12} \dots \mathbb{M}^{1k} \dots) \mathbf{X}^{\infty} = \sum_{k=1}^{\infty} \mathbb{M}^{1k} \mathbf{X}^{[k]}.$$

Уравнения (5–10) представляют собой суть матричного формализма для представления эволюции динамической системы (преобразования Ли) (3), описываемой системой обыкновенных дифференциальных уравнений (1). Подобные соотношения позволяют вычислять как базовые матрицы M^{1k} , так и вспомогательные M^{ik} , $i \leq k$ как в символьной, так и в численной моде.

3. Самосогласованная эволюция пучка

В предположении, что влиянием собственного заряда пучка можно пренебречь, приведенный в предыдущем разделе формализм позволяет вычислять блочные матрицы входящих в «матричный пропагатор» M^∞

$$M^\infty = \begin{pmatrix} M^{11} & M^{12} & \dots & M^{1k} & \dots \\ \mathbb{O} & M^{22} & \dots & M^{2k} & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbb{O} & \mathbb{O} & \dots & M^{kk} & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix},$$

Однако, в случае интенсивных пучков в уравнения движения входит слагаемое, которое определяет силы, генерируемые пучком частиц, что приводит к необходимости коррекции описанного выше метода. Действительно, в данном случае в процессе эволюции пучок генерирует собственное поле, что приводит к зависимости правой части уравнения (1) от состояния пучка как коллективного объекта. В этом случае уравнения движения принимают вид

$$(11) \quad \frac{d\mathbf{X}}{dt} = \mathbf{F}^{\text{ext}}(\mathbf{B}^{\text{ext}}(\mathbf{X}, t), \mathbf{E}^{\text{ext}}(\mathbf{X}, t), \mathbf{X}, t) + \mathbf{F}^{\text{self}}(\mathbf{B}^{\text{self}}(\mathbf{X}, t), \mathbf{E}^{\text{self}}(\mathbf{X}, t), \mathbf{X}, t),$$

где индекс «ext» обозначает внешнее (управляющее электромагнитное поле) и индекс «self» – собственное поле пучка соответственно. Именно второе слагаемое в процессе эволюции перестраивается в соответствии с динамикой пучка, в то время как первое слагаемое задает управляющее поле, которое должно обеспечить необходимый характер эволюции пучка. В этом случае операторное уравнение (2) необходимо переписать в виде самосогласованного уравнения

$$(12) \quad \frac{d\mathcal{M}(t | t_0; \mathcal{L}_{\mathbf{F}})}{dt} = \mathcal{L}_{\mathbf{F}} \circ \mathcal{M}(t | t_0; \mathcal{L}_{\mathbf{F}}), \quad \mathcal{L}_{\mathbf{F}}(\mathcal{M}) = \mathcal{L}_{\mathbf{F}}^{\text{ext}} + \mathcal{L}_{\mathbf{F}}^{\text{self}}(\mathcal{M}), \quad \mathcal{M}(t_0 | t_0; \mathcal{L}_{\mathbf{F}}) = \mathcal{I}d,$$

что приводит к необходимости применения самогласованной схемы вычисления соответствующего решения. Предполагая, что состояние пучка описывается функцией распределения частиц в фазовом пространстве $f(\mathbf{X}(t), t)$, можно предложить следующий алгоритм решения подобного уравнения. Записав уравнение (12) в интегральном виде мы получаем уравнение Вольтерры–Урысона II рода, так как фактически система: субъект управления (система транспортировки)–объект управления (пучок) охвачена обратной связью. Запишем уравнение (6) в следующем формальном виде:

$$(13) \quad \mathcal{M} = \mathcal{A} \circ \mathcal{M},$$

где \mathcal{A} – оператор Урысона II рода. Как известно, основным способом решения уравнений типа (13) является метод последовательных приближений, с помощью которого не

только доказывається существование неподвижной точки оператора \mathcal{A} , но и строится само решение, например, в соответствии со следующим алгоритмом.

Алгоритм.

Шаг 1. Выбирается функция распределения $f(\mathbf{X}(t), t_0) = f_0(\mathbf{X})$ из некоторого класса функций (см., например, [7]).

Шаг 2. По процедуре, описанной в предыдущем разделе, вычисляется оператор эволюции \mathcal{M} : $\mathcal{M}^0 = \mathcal{M}(t, | t_0; \mathcal{L}_{\mathbf{F}})$, $t \in [t_0, t_1]$.

Шаг 3. Вычисляется текущее значение функции распределения $f^1(\mathbf{X}, t) = f_0((\mathcal{M}^0)^{-1} \circ \mathbf{X}_0)$.

Шаг 4. Решаются соответствующие полевые уравнения с функцией распределения $f^1(\mathbf{X}, t)$ и находятся $(\mathbf{B}^{\text{self}})^1$, $(\mathbf{E}^{\text{self}})^1$.

Шаг 5. Вычисляются функция $(\mathbf{F}^{\text{self}})^1 = \mathbf{F}^{\text{self}}((\mathbf{B}^{\text{self}})^1, (\mathbf{E}^{\text{self}})^1, \mathbf{X}, t)$ или самосогласованный гамильтониан $(\mathcal{H}^{\text{self}})^1 = \mathcal{H}((\mathbf{B}^{\text{self}})^1, (\mathbf{E}^{\text{self}})^1, \mathbf{X}, t)$.

Шаг 6. Вычисляется $\mathcal{M}^1 = \mathcal{A} \circ \mathcal{M}^0$.

Шаг 7. Ищется новое значение $\langle f(\mathbf{X}, t_0) \rangle_{\mathfrak{M}^1}^1$, $0 < \alpha < 1$ по формуле

$$(14) \quad \langle f(\mathbf{X}, t_0) \rangle_{\mathfrak{M}^1}^1 = (1 - \alpha) \left\langle f_0 \left((\mathcal{M}^0)^{-1} \circ \mathbf{X}_0 \right) \right\rangle_{\mathfrak{M}^0} + \alpha \left\langle f_0 \left((\mathcal{M}^1)^{-1} \circ \mathbf{X}_0 \right) \right\rangle_{\mathfrak{M}^0},$$

Шаг 8. Проверяется критерий (или аналогичные критерии, адаптированные к конкретным задачам)

$$(15) \quad \|\mathcal{M}^k - \mathcal{A} \circ \mathcal{M}^{k-1}\| < \varepsilon, \quad k \geq 1.$$

Если условие (15) выполняется, то процесс определения оператора \mathcal{M} (и соответственно $f(\mathbf{X}, t)$, $\langle f(\mathbf{X}, t) \rangle_{\mathfrak{M}}$) на интервале $[t_0, t_1]$ заканчивается. Если (15) не выполняется, то процедура повторяется с шага 4 при соответствующем переопределении функций и операторов. Таким образом, по некоторому начальному значению \mathcal{M}^0 строится последовательность $\mathcal{M}^k = \mathcal{A} \circ \mathcal{M}^{k-1}$, доказывається, что последовательность \mathcal{M}^k сходится (в том или ином смысле) к некоторому «элементу» \mathcal{M}^* , и устанавливается равенство $\mathcal{M}^* = \mathcal{A} \circ \mathcal{M}^*$.

Предложенный алгоритм может быть реализован как в терминах функции распределения, так и в терминах так называемой матрицы огибающей (например, так называемой *rms*-огибающей (среднеквадратичной)) [7]. В обоих случаях мы описываем пучок также с помощью матриц.

Выше сказанное демонстрирует, что и в случае необходимости учета собственного заряда эволюция пучка описывается с помощью матриц \mathbb{M}^{ik} , только сама процедура вычисления несколько усложняется.

4. Дополнительные ограничения на матричное представление оператора эволюции

Следует отметить, что при реализации описанного формализма мы вынуждены при практических вычислениях обрывать ряд и обеспечивать выполнение соответствующих оценок точности вычислений. Обычно при подобной редукции соответствующих рядов проводят оценку соответствующих вычислительных погрешностей (влияющих на «количественную близость» приближенного и «точного» решений). Однако, для многих задач физики пучков (например, для задач длительной эволюции пучка частиц) необходимо с высокой степенью точности обеспечивать выполнение ряда дополнительных

свойств, например, условия симплектичности для гамильтоновых систем, закона сохранения энергии для консервативных систем и т. д. Невыполнение подобных условий приводит к существенному искажению качественных и количественных характеристик эволюции пучка. Именно поэтому при обрывании соответствующих рядов мы должны обеспечить выполнение соответствующих условий, накладывание которых приводит к дополнительным уравнениям связи между элементами матриц M^{1k} . Например, в случае гамильтоновых систем условие симплектичности приводит к линейным алгебраическим уравнениям с целыми коэффициентами на элементы матриц M^{1k} , $k \geq 2$. Следует заметить, что условие симплектичности является универсальным для всех гамильтоновых систем и может быть реализовано в общем виде (см. [7]). В то же время закон сохранения энергии для консервативных систем в каждом конкретном случае должно обеспечиваться специальным образом, поскольку для различных задач форма функции Гамильтона (полной энергии) различна, и из симплектичности (присущей всем гамильтоновым системам) не вытекает сохранение энергии (теорема Марсдена [9]) за исключением линейного приближения и небольшого числа частных задач. В частности, закон сохранения энергии приводит к необходимости введения поправочного параметра, выбор которого зависит от представления функции Гамильтона в том или ином случае. Соответствующие коррекции матричных элементов могут быть вычислены заранее и быть использованы по мере необходимости (см., например, [10]).

5. Концепция виртуального ускорителя

5.1. Введение

Система управления пучком реализуется с помощью внешнего электромагнитного поля, генерируемого управляющими элементами (свободными промежутками, диполями, мультиполями различного порядка). Структура управляющей системы выстраивается в соответствии с предназначением ускорительной установки и соответствующими критериями качества. Следует заметить, поскольку объект управления – пучок частиц содержит огромное число частиц, то использование в качестве функциональных критериев характеристик пучка приводит к существенным вычислительным затратам. Поэтому часто (по крайней мере на начальных этапах) в качестве функционалов рассматриваются некоторые упрощенные варианты. Использование же матричного формализма позволяет строить критерии качества в терминах условий на элементы матриц M^{1k} , что не только существенно снижает время вычислений, но и позволяет во многих случаях сводить соответствующие задачи оптимизации к решению функциональных уравнений для управляющих параметров. В случае плотных пучков или при необходимости управлять формированием распределения частиц в пучке кроме «эволюционных» матриц M^{1k} , необходимо ввести в рассмотрение матрицы, предназначенные для описания пучка как ансамбля частиц, например в виде матриц огибающих или в терминах функции распределения частиц в пучке [7].

Таким образом, основная задача физики пучков – синтез системы управления, обеспечивающей необходимые качества эволюции пучка (в том числе многооборотной), может быть формализован в терминах матриц, либо M^{1k} (в том числе для плотных пучков), либо в терминах вспомогательных матриц, сопоставленных основным классам задач. Данный подход позволяет рассматривать задачу построения оптимальной системы транспортировки пучка как задачу над семейством матриц, на котором соответствующие критерии формализованы в виде условий на матричные элементы. Именно это

обстоятельство и позволяет ввести в рассмотрение семейство унифицированных объектов – LEGO-объектов, представляющих собой либо матрицы, либо некоторые действия над соответствующими матрицами.

Следует отметить, что подобный подход позволяет создавать специальные базы данных и знаний соответствующих объектов и операций над ними. Это позволяет рассматривать задачу управления пучком частиц как задачу выполнения некоторых операций на множестве соответствующих матриц.

Очевидно, что для сложной системы управления, состоящей из огромного числа управляющих элементов и реализующей управление коллективным объектом – пучком частиц, на первый план выходит проблема снижения временных затрат. Это требование приводит к необходимости более тщательного отбора используемых объектов с точки зрения вычислительной эффективности вычислительных операций над ними.

5.2. Вычислительный кластер

Само определение Grid-системы предполагает наличие как вычислительных, так и вспомогательных ресурсов. В рамках проекта «Виртуальный ускоритель» на факультете ПМ-ПУ СПбГУ организовано два различных кластера. При необходимости каждый из них может рассматриваться как самостоятельный ресурс, однако при этом возрастают накладные расходы на пересылку информации между ними. Для данной схемы предложен достаточно дешевый, универсальный, хотя не обладающий достаточной производительностью вариант, использующий доступ к общим данным по протоколу NFS. В качестве системы хранения используется сетевой массив – Thecus N7700, представляющий из себя подключаемое к сети устройство, содержащее пять жестких диска WD 1 Тб. Диска собраны в RAID5-массив, что гарантирует сохранность данных в случае выхода из строя отдельного диска. Доступ к данным осуществляется по протоколу iSCSI, обеспечивающему достаточную производительность. Оба кластера в качестве коммуникационной среды используют сеть стандарта Gigabit Ethernet. Следует заметить, что проведенные тестовые вычисления с использованием сетей более низкой производительности не позволяли получать искомые решения за достаточно удовлетворительное время. Каждый кластер состоит из десяти вычислительных узлов и одного управляющего (мастер-хоста).

В качестве операционной системы рассматривались системы из семейства Linux, а в качестве дистрибутива рассматривались два варианта: Debian Linux и CentOS-5 (некоммерческая сборка Red Hat Enterprise Linux). Выбор первого дистрибутива во многом определен следующими преимуществами:

- достаточно представительный репозиторий, в котором можно найти большую часть необходимых программных пакетов;
- популярность и тем самым узнаваемость у пользователя;
- стабильность и устойчивость;
- «легковесность» – установка по сети занимает несколько минут, что существенно повышает масштабируемость кластера.

С другой стороны второй дистрибутив выбран с учетом следующих его преимуществ:

- гарантия устойчивой работы (все пакеты проверены на совместимость друг с другом);
- совместимость с рядом коммерческих приложений (например, с Ansys, Oracle,

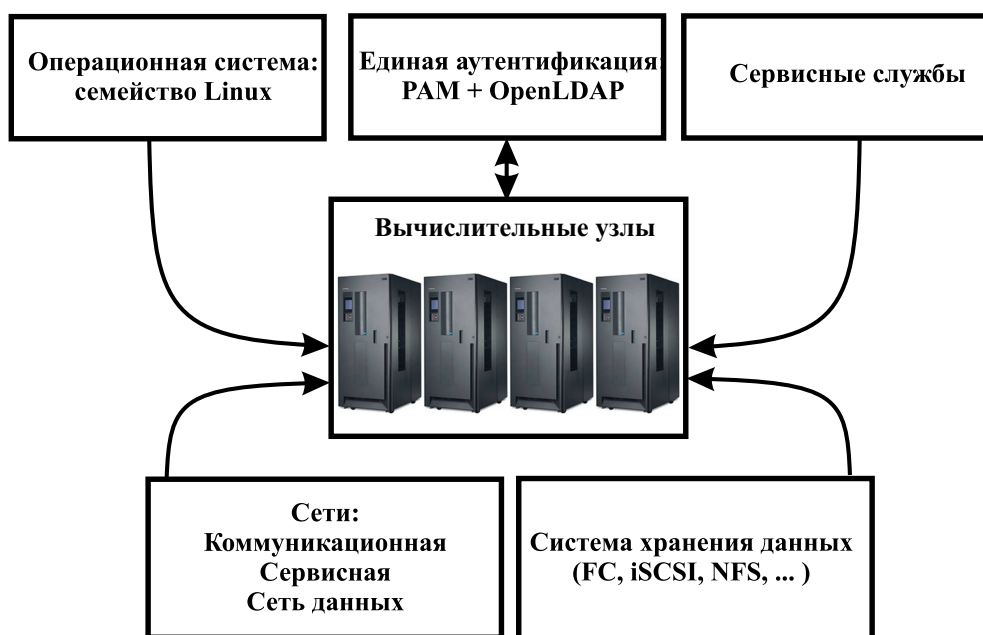


Рис. 1. Схема вычислительного кластера.

IBM DB2 и другими), что повышает возможности всей вычислительной инфраструктуры, поскольку подобные пакеты могут подключаться на заключительных этапах моделирования для построения более детальных (а значит и более ресурсоемких) моделей;

- встроенная поддержка кластерной системы хранения;

- «легковесность» – установка по сети занимает несколько минут, что существенно повышает масштабируемость кластера.

Поскольку изначально сам проект «Виртуальный ускоритель» предполагает, что кластер должен эксплуатироваться достаточно большим сообществом пользователей, то для хранения информации о запущенном задании (в том числе пользовательской информации) было реализовано централизованное хранилище данных аутентификации. Для реализации этой задачи достаточно традиционным решением является использование протокола LDAP (Lightweight Directory Access Protocol), предоставляющий доступ к службе каталогов X.50. К достоинствам этого протокола можно отнести все необходимые операции для доступа к вычислительным ресурсам. При этом любое вносимое изменение оказывает соответствующее влияние на каждый узел, вне зависимости от числа узлов. Для реализации вычислительного кластера из данного набора вычислительных узлов необходимо использовать интерфейс, обеспечивающий объединение все инфраструктуры в единый вычислительный кластер. В нашем случае мы использовали достаточно хорошо апробированный интерфейс MPI (Message Passing Interface), представленный библиотекой MPICH (см. рис. 2). Для обновления нам нет необходимости устанавливать отдельную копию пакета на каждый узел. Построение GRID системы осуществлялось в соответствии со схемой, представленной на Рис. 3. В качестве промежуточного программного обеспечения для построения GRID выбран инструментарий Globus Toolkit. Одной из причин выбора данного инструментария является полное соответствие стандарту Open Grid Services Architecture (OGSA), предусматривающий встраивание в GRID-системы средств поддержки языка XML и будущих стандартов Web-сервисов. Кроме того, данный инструментарий не только популярен в академической среде, что сокращает время ознакомления сотрудников с новыми для них интер-

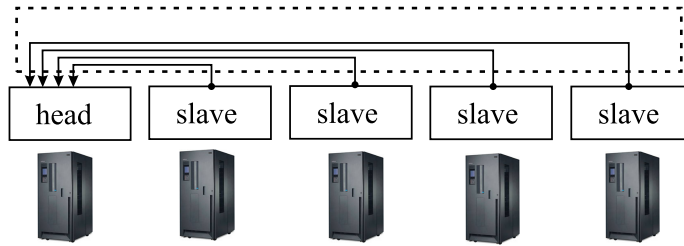


Рис. 2. Параллельная среда MPICH.

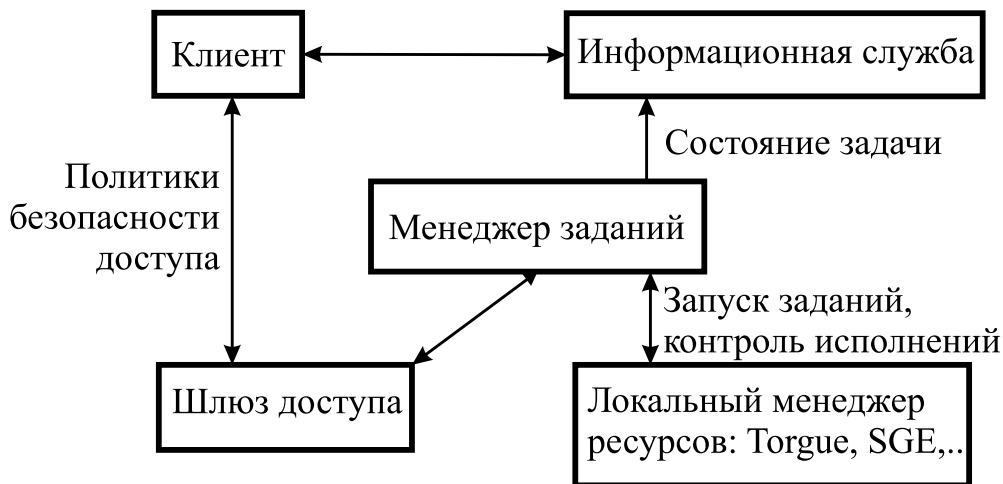


Рис. 3. Схема окружения GRID.

фейсами разработки приложений, но и поддерживается крупными ИТ-компаниями.

Локальный менеджер ресурсов, устанавливаемый на каждом кластере должен обеспечивать выполнение следующих задач.

- 1) Получение исчерпывающей информации по вычислительным узлам и иметь над ними полный контроль.
- 2) Ставить параллельные задания пользователей в очередь на исполнения и выполнять их в соответствии с заданными приоритетами.
- 3) Предоставление пользователям необходимой информации по заданиям и контроля за ними (временная остановка, удаление и пр.).
- 4) Обеспечение безопасности данных каждого пользователя.

В рамках данного проекта рассматривались два менеджера ресурсов Torgue и Sun Grid Engine. Последний ресурс интенсивно разрабатывается, поэтому выбран именно он (<http://gridengine.sunsource.net/>). Используемые узлы следует разделить на следующие типы:

Master Host. Управляющий узел, координирующий работу всего кластера (ведет учет заданий пользователей, следит за очередями и обеспечивает безопасность работы).

Shadow Master Host. Резервный управляющий узел, однако, в нашем случае не используется, так как в настоящее время число узлов невелико.

Administration Host. Обеспечивает перенастройку кластера в связи с политиками безопасности.

Submit Host. Узел, обеспечивающий постановку пользовательских заданий в очередь.

Execution Host. Вычислительный узел, имеющий определенное число слотов, задающих число одновременно запущенных задач.

После развертывания локальных менеджеров ресурсов на каждом кластере устанавливается инструментарий Globus Toolkit.

Для пересылки данных между кластерами в пределах GRID окружения был использован сервис Globus Reliable File Transfer, для работы которого использована сетевая база данных (в частности, PostgreSQL). Развернутый инструментарий Globus Toolkit необходимо интегрировать с локальным менеджером ресурсов. Для этой цели был использован Globus Gram WS JobManager, совместимый с используемым SGE.

6. Практические задачи

6.1. Поиск динамической апертуры

В качестве одного из примеров можно рассмотреть задачу поиска так называемой динамической апертуры (ДА) ускорителя (обычно циклического) \mathfrak{A} , обладающей следующими свойствами:

1) Пусть для некоторого значения продольной координаты $s_0, \forall x_0, y_0 \in \mathfrak{A}$, тогда для любого достаточно большого $N > 1$ $x(s_0 + N \cdot T), y(s_0 + N \cdot T) \in \mathfrak{A}$, где T – период ускорителя. Иными словами, данное множество через каждый оборот переходит само в себя, тем самым обеспечивая динамическую устойчивость эволюции пучка в ускорительном канале.

2) Для любой начальной частицы с пространственными координатами $\forall x_0, y_0 \notin \mathfrak{A}$ ее образ удовлетворяет условию $x(s_0 + N), y(s_0 + N) \notin \mathfrak{A}$, т. е. ее «образ» выходит за апертуру ускорителя при достаточно большом N .

Понятие динамической апертуры играет большую роль в теории и практике ускорительной физики. Ее поиск и максимизация по площади является одной из важнейших задач. Для этой цели в базовую структуру ускорителя добавляются нелинейные управляющие элементы, компенсирующие нелинейности, присущие всем реальным управляющим элементам.

Для поиска динамической апертуры, обычно имеющей достаточно сложную по форме границу, в работе рассмотрен метод лучей. Согласно этому методу на каждом луче (в поперечном конфигурационном пространстве) по некоторому алгоритму ищется «крайняя» точка, не удовлетворяющая первому условию (см. рис. 4). При этом распа-

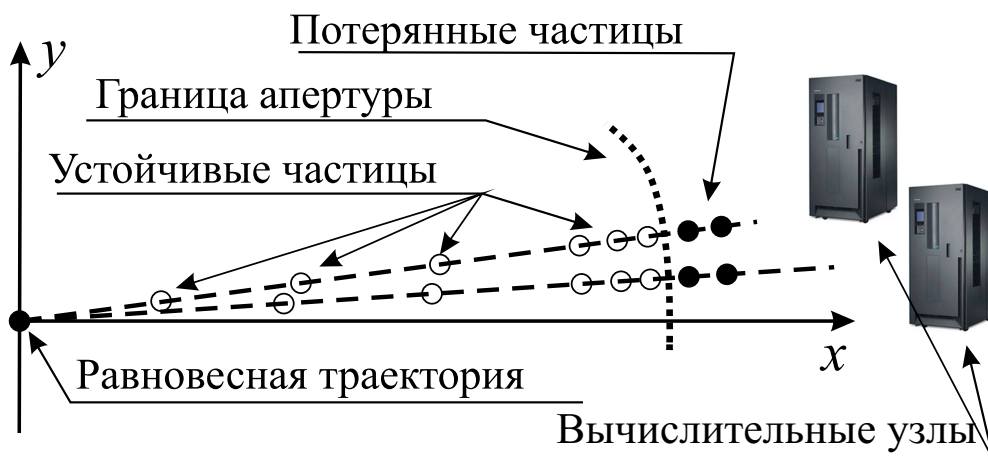


Рис. 4. Схема окружения GRID.

раллеливание первого уровня осуществляется именно по лучам. Иными словами каждый луч является самостоятельным процессом, который, в свою очередь, также распараллеливается (группировка по M точек). В качестве параллельных потоков третьего уровня можно рассмотреть матричные умножения.

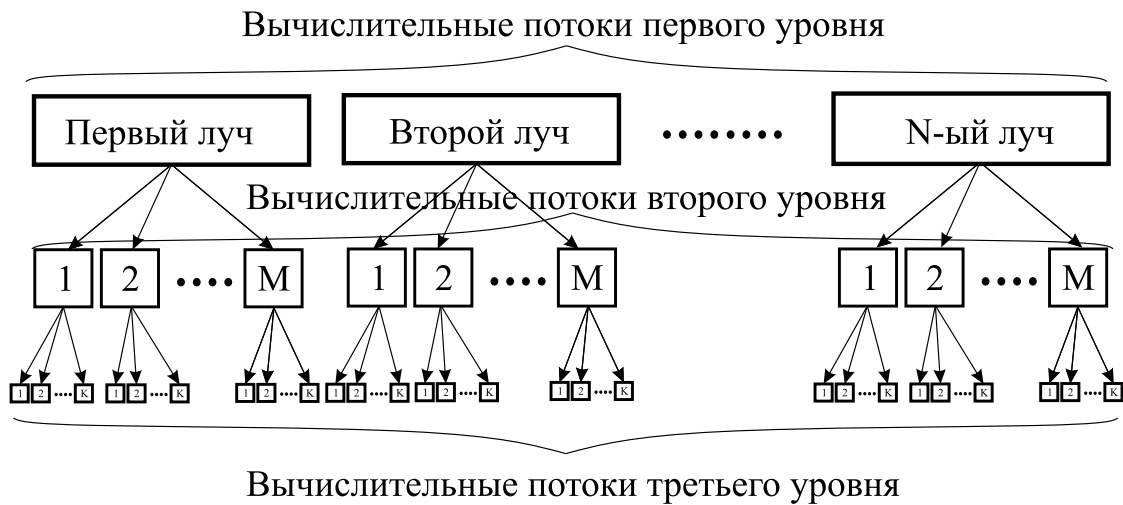


Рис. 5. Схема окружения GRID.

В данном процессе распараллеливание осуществлялось по целому ряду математических операций (см. рис. 5):

- распараллеливание по лучам (N лучей);
- распараллеливание по частицам, принадлежащим одному лучу – двухуровневое распараллеливание: общее число частиц M делится на группы по K частиц;
- распараллеливание матричных вычислений.

Описанный алгоритм можно записать в форме следующего математического выражения (см. второй раздел).

$$(16) \quad \mathbb{X}^k = \sum_{j=1}^J \mathbb{M}^{1j}(s|s_0)(\mathbb{X}_0^k)^{[j]}, \quad \dim \mathbb{X}^k = \dim \mathbb{X}_0^k = 2n \times M, \quad k = \overline{1, N}$$

где J – порядок нелинейностей, рассматриваемых в задаче (для современных ускорителей $3 \leq J \leq 5$), $N \cdot M$ – полное число частиц, рассматриваемых в задаче, и $\mathbb{X}^j = (\mathbf{X}_1^j, \dots, \mathbf{X}_M^j)$, $\forall j = \overline{1, N}$, – матрица, состоящая из фазовых векторов \mathbf{X}_i , $i = \overline{1, M}$, принадлежащих j -му лучу. Заметим, что в данном случае распараллеливание является практически идеальным, поскольку эволюция каждой частицы не зависит эволюции остальных.

К данному классу задач, в частности относится задача медленного вывода частиц из циклического ускорителя, также требующая ресурсоемких вычислений [7]. Для данного класса задач особое значение имеет информация о степени влияния различных нежелательных эффектах (например, токовые пульсации) на качество выводимого пучка. Поскольку подобный вывод осуществляется на протяжении $5 \cdot 10^5 - 10^6$ оборотов, то проблема повышения вычислительной эффективности соответствующих алгоритмов чрезвычайно актуальна. Как и в предыдущей задаче, взаимодействием частиц между

собой можно пренебречь (по крайней мере, на первых этапах). В этом случае динамика частиц, как и в предыдущей задаче, описывается уравнениями типа (??). Однако в данном случае для осуществления вывода частиц необходимо адиабатически изменять характеристики управляющего магнитного поля. Именно адиабатичность позволяет использовать алгоритм вычисления результирующих матричных отображений, основанный на итеративном возведении в степень матриц, ответственных за однооборотную эволюцию. Именно данный подход позволил существенно сократить время вычислений и получить интересные и адекватные экспериментальным данным результаты. Данный подход в дальнейшем предполагается использовать при создании системы контроля управления медленным выводом частиц из нуклотрона в рамках Международного проекта NICA [11].

6.2. Динамика частиц с собственным зарядом

В качестве примера задачи, в которой задача распараллеливания процессов не столь тривиальна, можно привести задачу транспортировки интенсивного пучка в ускорительном канале. В этом случае взаимодействие частиц нарушает независимость эволюции частиц. В соответствии с алгоритмом, изложенным в третьем разделе, первые три шага могут быть распараллелены, поскольку на этих этапах именно матричный формализм в полной мере применим (практически аналогично технологии распараллеливания в задаче поиска динамической апертуры). Критическими становятся четвертый и пятый шаги, на которых обычно (в стандартных алгоритмах) осуществляется перекрестный обмен информацией между вычислительными потоками. В соответствии с предложенным алгоритмом (подробности изложены в ряде работ автора и достаточно подробно освещены в книге [7]) происходит расщепление вычислительных потоков. Естественно, что появляется необходимость совершать дополнительные операции на 6–8 шагах, обеспечивая выполнение условия (15). Иными словами, мы «развязали» сильную связанность взаимодействия частиц друг с другом заменив ее новой парадигмой решения самосогласованных задач. В соответствии с этой парадигмой, на каждом шаге частицы пучке при переходе от одной точки на опорной траектории поле не пересчитывается синхронно с эволюцией пучка. Изменение происходит при вычислении виртуального распределения (см. шаг 7 алгоритма), что затем учитывается при возврате к четвертому шагу алгоритма в случае невыполнения условия (15). Естественно, что данный подход имеет свои ограничения, как и достаточно традиционные модели, например, Particle-in-Cell [12] или модель взаимодействия каждой частицы с каждой (см., например, [13]). Несмотря на то, что данный подход продемонстрировал свою адекватность при решении ряда практических задач, он естественно может быть применим только при определенных условиях. Однако, именно возможность естественного параллелизма (именно в силу использования матричного формализма) позволяет применять его в рамках концепции «Виртуальный ускоритель», поскольку он обеспечивает необходимую производительность и точность. После проведения тестовых вычислений с выбором некоторого набора контрольных параметров управляющей системы необходимо подтвердить (или опровергнуть) корректность выбора соответствующего набора управляющих параметров. Данный подход, конечно, имеет четко ограниченные границы применимости, однако во многих задачах, в качестве базового инструментария для проведения «быстрых вычислений» он в полной мере себя оправдывает, поскольку допускает естественное распараллеливание (и распределение) соответствующих вычислительных потоков. Напомним, что одним из предназначений концепции «Виртуальный

ускоритель» является проведение эффективных (прежде всего по времени) вычислительных экспериментов для получения набора управляющих параметров, необходимых для настройки системы управления. Особо необходимо отметить, что данное достоинство особо ценно как при быстрой переналадке системы питания управляющими элементами, так и при подборе семейства параметров, в достаточной степени удовлетворительных для обеспечения работоспособности системы, но все-таки требующих уточнения в случае выбора какого-либо набора данных параметров из полученного семейства. На этом этапе можно проводить более масштабные вычислительные эксперименты с применением и других моделей и программных решений для уточнения управляющих параметров в соответствии с формализованными критериями оптимальности системы.

7. Заключение

Рассмотренные в статье методы и технологии позволили реализовать некоторые идеи, заложенные в концепцию «Виртуальный ускоритель». Предложенные математические модели и соответствующие алгоритмы продемонстрировали несомненную эффективность и корректность при описании рассматриваемой предметной области. В дальнейшем предполагается развитие описанного подхода как по наполнению необходимыми для данной предметной области математическими моделями, реализующими их алгоритмами и программными решениями, допускающими распараллеливание и распределение вычислительных потоков, в том числе и с использованием кластерных решений на графических картах. Дальнейшее развитие данной работы предполагает создание полигона как для обучения студентов и аспирантов в области моделирования ускорительных систем, так и в области моделирования сложных информационных систем, позволяющих проводить эффективные вычислительные эксперименты, накапливать и хранить необходимые данные и использовать их при решении практических задач. Особое значение в данном направлении будут играть такие инструменты и технологии искусственного интеллекта, как средства компьютерной алгебры, теории и практики баз данных и знаний, искусственного интеллекта и т. п. (см., например, [14]) Именно использование экспертных систем в системах контроля работы ускорителей предъявляет особые ограничения на качество, достоверность данных, получаемых за приемлемое время. Специфика подобного рода систем накладывает определенные ограничения на используемые объекты. В частности, именно однородность, достаточно хорошая формализация позволяет строить соответствующие правила отбора «правильных» решений, которые могут быть положены в основу алгоритма управления реальной системой питания ускорительными системами. Описанные в данной статье результаты и технологии указывают на достаточную их совместимость с концепцией «Виртуального ускорителя». Эффективность предлагаемых подходов продемонстрирована рядом проведенных вычислительных экспериментов, небольшая часть из которых описана в статье.

Автор работы хотел бы высказать слова благодарности своим сотрудникам, студентам и аспирантам, за оказание большой помощи в реализации и верификации предлагаемых идей. Прежде всего, я хотел бы упомянуть В.В.Корхова, А.Н.Иванова, Д.В.Зюзина, Н.В.Кулабухову и А.В.Лазарева.

СПИСОК ЛИТЕРАТУРЫ

1. Chiu P.C., Kuo C.H., Chen J. et al. Virtual Accelerator Development for the TPS. Proc. of IPAC'10, Kyoto, Japan, P. 2728–2730.
2. EPICS Home Page. <http://www.aps.anl.gov/epics/>
3. *Andrianov S.N.* LEGO-Technology Approach for Beam Line Design // Proc. of EPAC 2002. Paris. France. P. 1607–1609.
4. *Dragt A.J.* Lectures on Nonlinear Orbit Dynamics. Physics of High Energy Particle Accelerators / AIP Conf. Proc. N.-Y., 1982. N 87. P. 147–313.
5. *Аграчев А.А., Гамкредидзе Р.В.* Экспоненциальное представление потоков и хронологическое исчисление // Мат. сборник. 1978. Т. 107. № 4(12). С. 467–532.
6. *Magnuss W.* On the Exponential Solution of Differential Equations for a Linear Operator. Comm. Pure Appl. Math. 1954. Vol. 7. No 4. P. 649–679.
7. *Андрюанов С.Н.* Динамическое моделирование систем управления пучками частиц. СПб. СПбГУ, 2004.
8. *Magnuss W.* On the Exponential Solution of Differential Equations for a Linear Operator. Comm. Pure Appl. Math. 1954. Vol. 7. No 4. P. 649–679.
9. G. Zhong, J. E. Marsden, Lie-Poisson Hamilton-Jacobi theory and Lie-Poisson. Physics Letters. V. 133. No. 3. 1988. P. 134–139.
10. Ivanov A., Andrianov S. Matrix Formalism for Long-Term Evolution of Charged Particles and Spin Dynamics in Electrostatic Fields. Will be appear in Proc. of Intern. Computational Accelerator Conf. 19–25 Aug. 2012. Rostock. Germany.
11. <http://nica.jinr.ru/>
12. Hatzky R., Bottino A. Particle in Cell Methods in Plasma Physics. Summer School on P3C challenges in Computational Sciences, 4–7 Oct. 2010, Acireale. 40 P.
13. Giovannozzi M. Space-Charge Simulations Using Parallel Algorithms. Proc. of Sixth European Particle Accelerator Conference (EPAC'98). 22–26 June 1998, Stockholm. P. 1189–1191.
14. Klein W.B., Westervelt R.T., Luge G.L. Developing a General Purpose Intelligent Control System for Particle Accelerators. J. of Intelligent and Fuzzy Systems, Vol. 7, No 1. 1999. 12 P.