

Модель проблемно-ориентированной облачной вычислительной среды

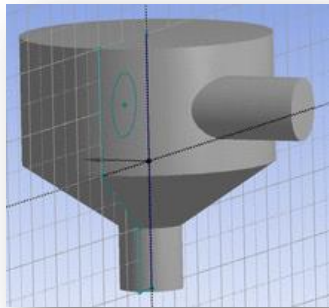
РАДЧЕНКО Г.И., К.Ф.-М.Н., ДОЦЕНТ КАФ. СП.

ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

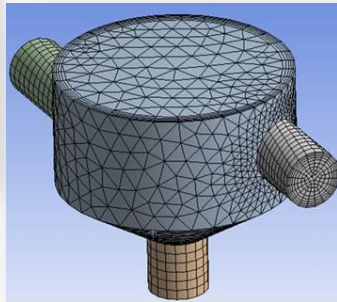
Проблематика и цель работы

- Современным задачам вычислительной науки характерны высокие требования к предоставляемым вычислительным ресурсам, а также сложная вычислительная структура заданий, которую можно описать в виде потока работ.
- Также, для задач такого типа характерны многовариантные расчеты, когда вычислительное задание запускается сотни и тысячи раз с различными вариациями входных параметров.
- *Цель работы:* разработать методы и алгоритмы планирования потоковых приложений для решения задач вычислительной науки в облачных вычислительных системах

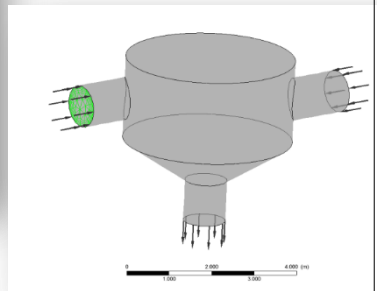
Пример потока работ для задачи моделирования миксера



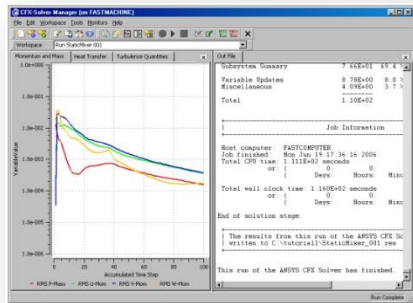
Design Modeler



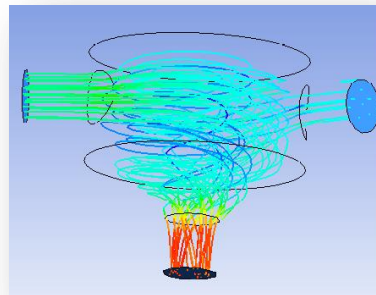
CFX-Mesh



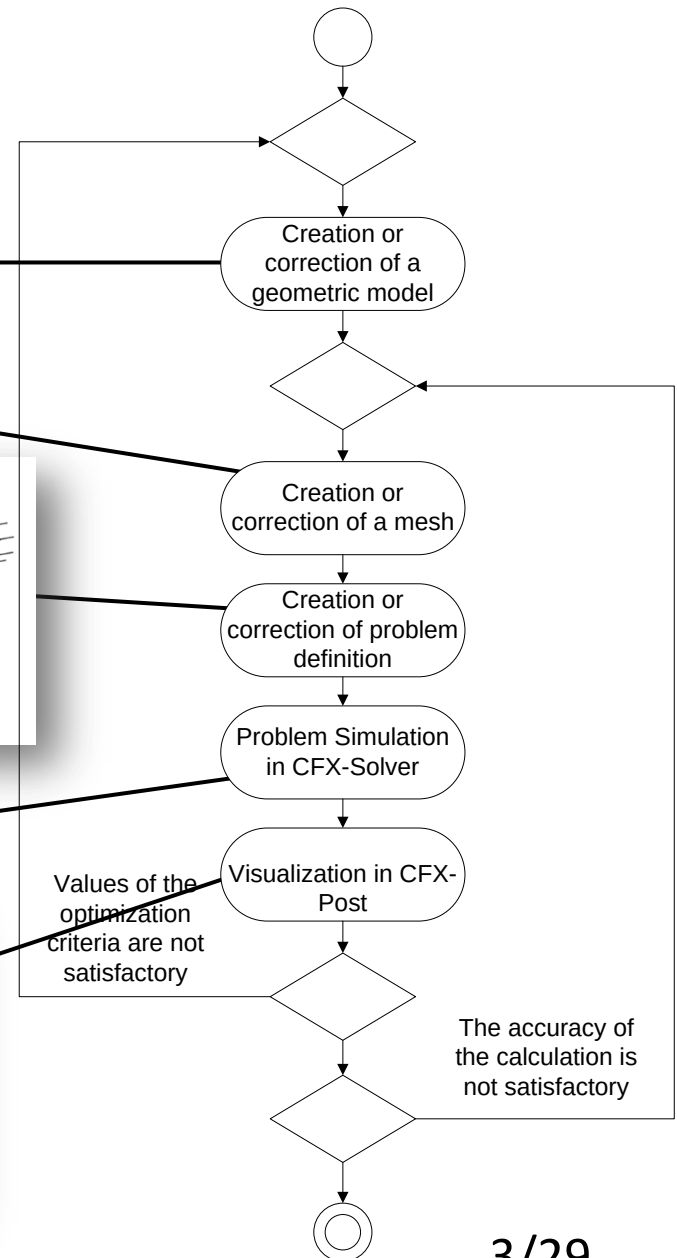
CFX-Pre



CFX-Solver



CFX-Post



Предметная область в ПВС

- Особенностью распределенных проблемно-ориентированных вычислительных сред (ПВС) является то, что они обеспечивают решение заданий в рамках **конкретной предметной области**.
- С одной стороны, это накладывает ограничения на классы задач, которые могут быть решены в рамках ПВС.
- С другой стороны, такое ограничение позволяет использовать **информацию о предметной области для прогнозирования вычислительных характеристик задач** при планировании и распределении заданий, увеличивая эффективность использования доступных вычислительных ресурсов.

Определение предметной области

Информационный объект

*Множество базовых информационных объектов $\mathcal{B}$ – это счетное множество объектов, которые в рамках соответствующей предметной области рассматриваются как атомарные (т.е. не имеющие составных частей), и разбиваются на k множеств *непересекающихся базовых классов* B_i :*

$$\mathcal{B} = \bigcup_{i=1}^k B_i, (\forall i \neq j) B_i \cap B_j = \emptyset.$$

Пример базовых классов информационных объектов:

- строки;
- целые числа;
- булевы переменные.

Класс информационных объектов

Определим *класс информационных объектов* C следующим образом:

- базовый класс информационных объектов будет являться классом информационных объектов:

$$C = (B), \forall B \in \mathcal{B};$$

- упорядоченный набор базовых классов информационных объектов будет являться классом информационных объектов:

$$C = (B_1, \dots, B_m), (\forall l \in \{1, \dots, m\}) B_l \in \mathcal{B};$$

- упорядоченный набор ранее определенных классов информационных объектов так же будет являться классом информационных объектов:

$$C = (C_1, \dots, C_u).$$

Информационный объект

Таким образом, информационный объект I класса $C = (C_1, \dots, C_u)$ представляет собой упорядоченный набор конечного числа информационных объектов $I = (I_1, \dots, I_u)$

Примером информационного объекта может быть:

- 3d модель физического процесса;
- Расчётная сетка;
- Результаты моделирования.

Множество классов информационных объектов

Определим *множество классов информационных объектов* $\mathcal{C}$ как конечное множество всех определенных в рамках предметной области классов информационных объектов:

$$\mathcal{C} = \bigcup_{i=1}^{|\mathcal{C}|} C_i .$$

Набор функций над $\mathcal{C}$

Определим набор функций $\mathcal{F}$ над множеством $\mathcal{C}$:

$$\mathcal{F} = \bigcup_{i=1}^{|\mathcal{F}|} f_i, \text{ где } (\forall i \in \{1, \dots, |\mathcal{F}|\}) f_i: \mathcal{C}_i^{in} \rightarrow \mathcal{C}_i^{out}$$

Каждая функция $f \in \mathcal{F}$ на вход получает n информационных объектов $\mathcal{J}^{in} = (I_1^{in}, \dots, I_n^{in})$. Результатом работы функции являются m новых информационных объектов $\mathcal{J}^{out} = (I_1^{out}, \dots, I_m^{out})$ соответствующих классов $\mathcal{C}^{out} = (C'_1, \dots, C'_m)$.

Примеры таких функций:

- функция преобразования расчётной сетки из одного формата в другой;
- функция генерации сетки из определенной геометрии;
- функция визуализации результатов моделирования.

Определение предметной области

Определим *предметную область* $\mathfrak{P}$ как упорядоченную тройку:

$$\mathfrak{P} = \langle \mathcal{B}, \mathcal{C}, \mathcal{F} \rangle$$

где

- $\mathcal{B}$ - множество базовых информационных объектов,
- $\mathcal{C}$ - множество классов информационных объектов,
- $\mathcal{F}$ – множество функций, определенных над $\mathcal{C}$.

Проблемно-ориентированная облачная среда

Проблемно-ориентированная облачная вычислительная среда

Определим *проблемно-ориентированную облачную вычислительную среду* как четверку:

$$\mathcal{C}(\mathcal{N}, \mathcal{E}, \mathcal{M}, \mathcal{P}),$$

где

- $\mathcal{N}$ – множество узлов вычислительной системы;
- $\mathcal{E}$ – множество сетевых соединений, связывающих вычислительные узлы;
- $\mathcal{M}$ – множество образов виртуальных вычислительных машин, доступных для развертывания на узлы из множества $\mathcal{N}$,
- $\mathcal{P}$ – базовая предметная область

Вычислительный узел

Вычислительным узлом $n \in \mathfrak{N}$ назовем вычислительную систему с общей памятью, представленную тройкой:

$$(P_n, m_n, \Pi_n),$$

где

- P_n – это упорядоченное множество вычислительных ядер узла;
- m_n – это объем оперативной памяти, доступной на вычислительном узле;
- Π_n – это вектор характеристик производительности вычислительного узла.

Образ виртуальной машины

Образом виртуальной машины $m \in \mathfrak{M}$ назовем тройку:

$$(P_m, m_m, \Pi_m),$$

где

- P_m – это упорядоченное множество вычислительных ядер, выделенных виртуальной машине;
- m_m – это объем оперативной памяти, выделенный виртуальной машине;
- Π_m – это вектор характеристик производительности виртуальной машины.

Характеристика производительности

Определим характеристику производительности вычислительной машины как отображение:

$$\pi: \mathfrak{M} \rightarrow \mathbb{Z}_{>0},$$

где $\mathfrak{M}$ – образ это виртуальной машины (либо вычислительный узел) существующий в вычислительной системе $\mathfrak{S}$.

Примерами характеристик производительности могут служить числовые характеристики машины, результаты синтетических тестов (Dhrystone, SuperPI, LINPACK, LAPACK и др.) или результаты тестового выполнения конкретных классов функций с заранее определенными наборами входных данных.

Всем машинам присвоим по вектору из r таких характеристик:

$$\Pi: \mathfrak{M} \rightarrow \mathbb{Z}_{>0}^r$$

Операторы проблемно-ориентированной среды

Будем требовать, чтобы в рамках проблемно-ориентированной вычислительной среды, для каждого класса задач были определены следующие функции для прогноза процесса выполнения задачи в зависимости от значений входных параметров:

- функция оценки объема выходных данных при определенных входных параметрах;
- функция оценки времени выполнения задачи при определенных входных параметрах на машине с указанным вектором характеристик производительности.

Оператор ожидаемого выхода

Для каждой функции $f \in F$ из предметной области $\mathfrak{F}$, выполняющейся в проблемно-ориентированной среде $\mathfrak{S}$, определим

оператор ожидаемого выхода $\nu(f, \mathcal{J}^{in})$ – это оператор, возвращающий ожидаемый общий размер в байтах всех выходных информационных объектов $\mathcal{J}^{out}$:

$$\nu(f, \mathcal{J}^{in}) = |\mathcal{J}^{out}| = \sum_{\forall I \in \mathcal{J}^{out}} \sum_{\forall b \in I} |b|.$$

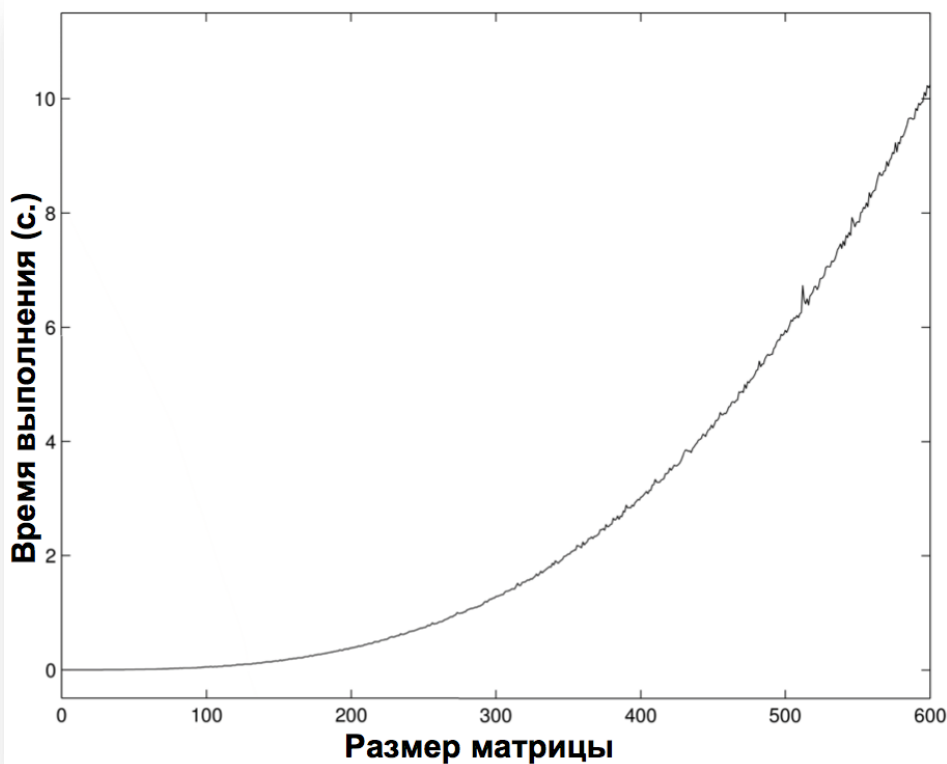
Оператор ожидаемого времени выполнения

Также, для каждой функции $f \in F$ определим *оператор ожидаемого времени выполнения функции* $\tau(f, \mathcal{J}^{in}, \Pi)$, возвращающий оценочное время выполнения (в секундах) функции f при заданном множестве входных информационных объектов $\mathcal{J}^{in}$ на машине, с вектором характеристик производительности Π :

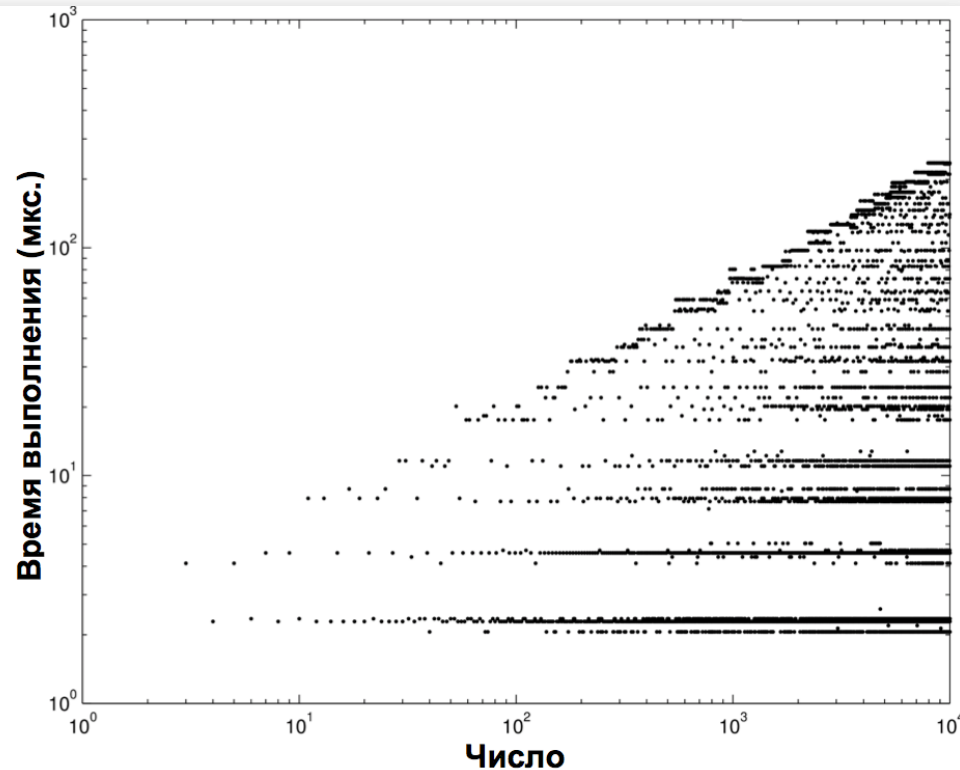
$$\tau: (f, \Pi, \mathcal{J}^{in}) \rightarrow \mathbb{N}.$$

Характеристики выполнения функций как случайные величины

Время выполнения — как случайная величина



Время выполнения алгоритма Холецкого строго зависит от размера входной матрицы



Время выполнения алгоритма перебора делителей слабо коррелирует со значением входного числа

Время выполнения – как случайная величина

Оценку времени выполнения функции можно смоделировать в виде случайной величины:

$$\chi(f, \Pi, \mathcal{J}^{in}) = \tau(f, \Pi, \mathcal{J}^{in}) + \alpha,$$

где

$\tau(f, \Pi, \mathcal{J}^{in})$ – детерминированная функция, представляющая зависимость времени выполнения функции f на машине с вектором характеристик производительности Π от вектора входных информационных объектов $\mathcal{J}^{in}$,

α – это стохастическая величина с нулевым математическим ожиданием ($M[\alpha] = 0$), представляющая факторы, не входящие в разрабатываемую модель.

Оценка времени выполнения

Если функция f выполняется на машине, с определенными характеристиками производительности Π_0 , то для оценки времени выполнения, можно использовать алгоритм k-ближайших соседей:

$$\hat{E}[\chi(f, \Pi_0, j^{in})] = \hat{t}(f, \Pi_0, j^{in}) = \frac{1}{k} \sum_{i=1}^k W_i(j^{in}) t_{j_i^{in}, \Pi_0}^f$$

где

$t_{j_i^{in}, \Pi_0}^f$ - предыдущее наблюдаемое время выполнения функции f с входными параметрами j_i^{in} на машине с характеристиками производительности Π_0 .

$W_i(j^{in})$ – весовая функция, назначающая тем большие веса значениям времени выполнения, чем ближе параметры запуска к значению j^{in} .

Учет производительности виртуальной машины

Для учета вектора производительности виртуальной машины, можно расширить понятия «входные параметры» функции, включив в них и параметры машины, на которой функция выполняется:

$$P = [I_1^{in}, \dots, I_n^{in}, \pi_1, \dots, \pi_r].$$

$$\hat{E}[\chi(f, \Pi, j^{in})] = \hat{\tau}(f, \Pi, j^{in}) = \frac{1}{k} \sum_{i=1}^k W_i(P) t_{P_i}^f$$

где

$t_{P_i}^f$ - предыдущее наблюдаемое время время выполнения функции f с параметрами запуска P_i

Оценка объема выходных данных

Для оценки объема выходных данных, применяя аналогичный подход, можно вывести следующую формулу оценки:

$$\hat{v}(f, \mathcal{J}^{in}) = \frac{1}{k} \sum_{i=1}^k W_i(\mathcal{J}^{in}) v_{\mathcal{J}_i^{in}}^f,$$

где

$\hat{v}(f, \mathcal{J}^{in})$ - это оценочное значение объема выходных данных функции f на базе информации об объеме выходных значений k предыдущих запусков той же функции с входными параметрами $\mathcal{J}_i^{in}$, и весовой функции W_i

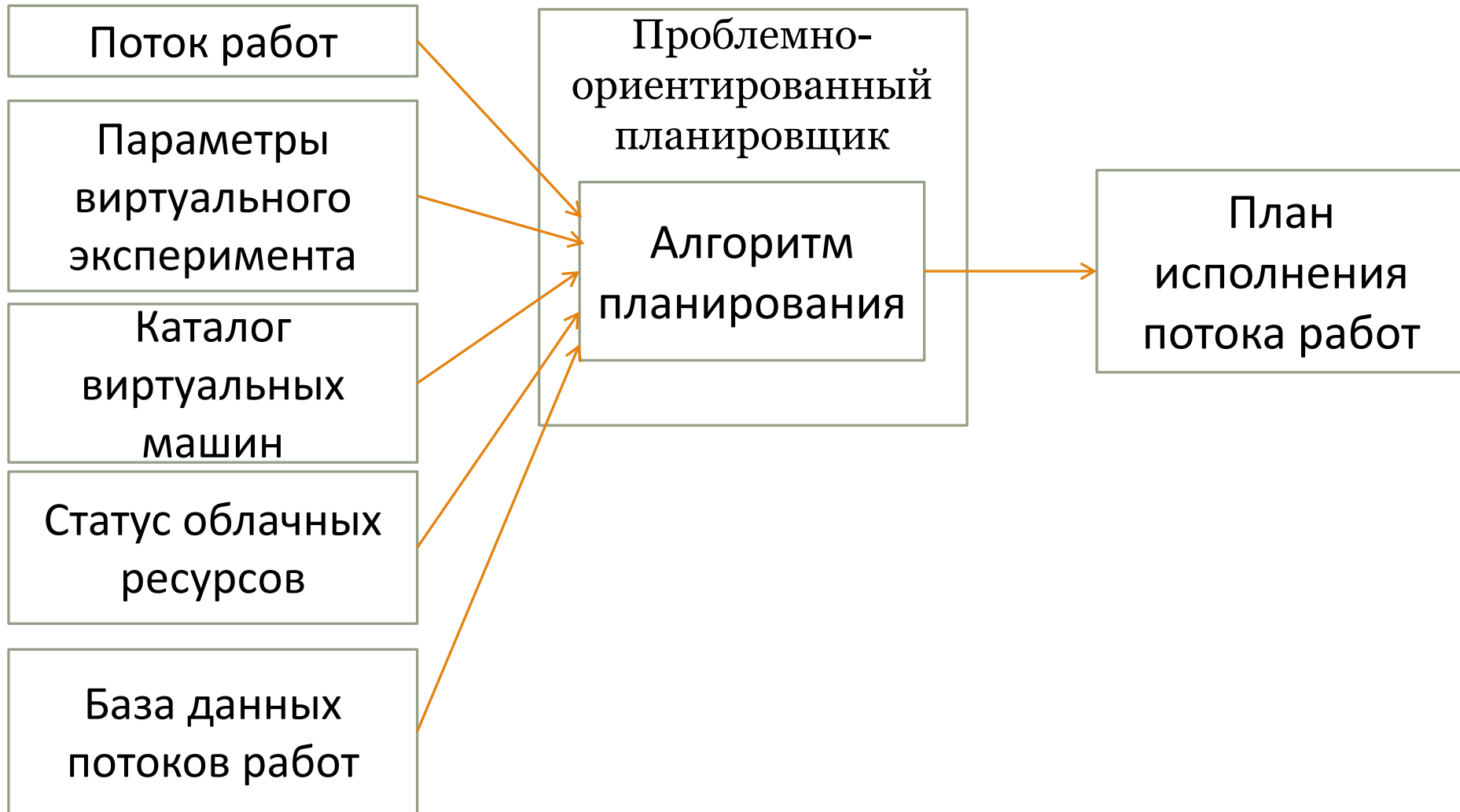
Применение модели для планирования потоков работ

Проблемно-ориентированный планировщик

Для реализации проблемно-ориентированной облачной вычислительной среды, разрабатывается проблемно-ориентированный планировщик потоков работ, включающий в себя:

- Базу данных с информацией о запусках потоков работ
- Систему прогнозирования характеристик выполнения задач в облачной вычислительной среде (как отдельных задач, так и потоков работ в целом)
- Систему планирования потоков работ на облачные вычислительные ресурсы, включая распределение задач по виртуальным машинам и распределение виртуальных машин по вычислительным узлам
- Систему мониторинга характеристик и перепланирования выполняющихся потоков работ

Проблемно-ориентированный планировщик



Задачи которые необходимо решить

Задачи:

- Создание базы данных для прогнозирования времени выполнения задач и расчет оценки времени выполнения задач на основе данной базы.
- Разработка математической модели потоковых приложений при решении задач инженерного анализа в распределенных вычислительных средах.
- Разработка алгоритма планирования потоковых приложений с эффективным для пользователя выделением виртуальных машин для решения его приложения; возможностью динамической корректировки плана исполнения приложения; поддержкой многовариантности приложений.
- Создание экспериментального образца программного обеспечения «Менеджер ресурсов»