Динамическое управление ресурсами на узлах в гетерогенной среде PanDA (на примере HPC @ NRC KI)

Лаборатория технологий больших данных для проектов в области Мега-сайенс

Новиков А.М.

05.2017

План доклада

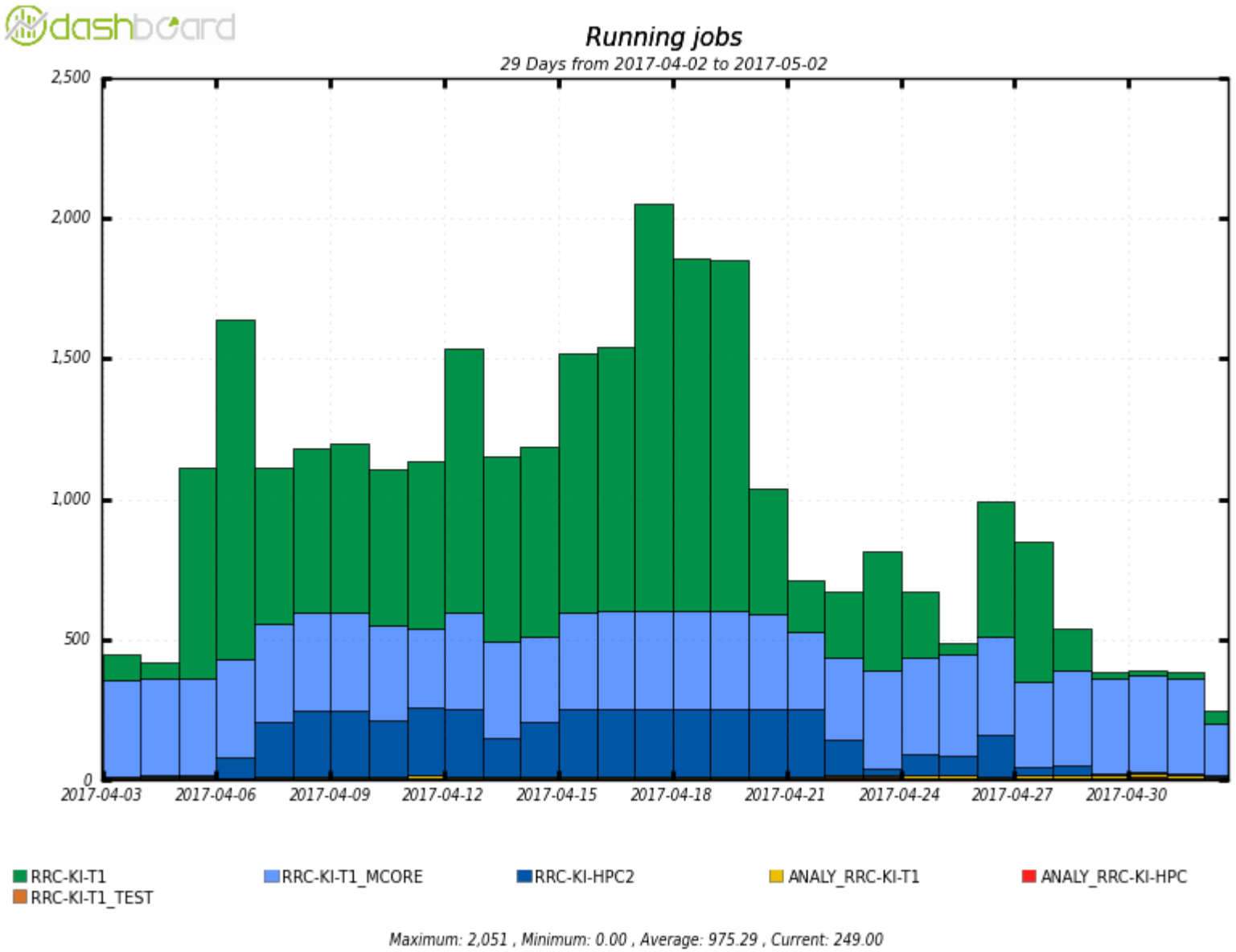
- Текущее состояние «PanDA at HPC KI»
- Задачи, требующие решения
- Возможные способы решения
- Обсуждение?

PanDA at NRC KI: Текущее состояние (1)

PanDA ресурсы (или очереди): однородные, HPC и Tier1(grid)

Параметр	ANALY_RRC-KI-HPC2	RRC-KI-HPC2	RRC-KI-T1	RRC-KI-T1_MCORE
узлов	**2**	32	~140	~140
Jobs/node, RAM/core	8 2	8 2	8 2	3-6 (corecount=8) **16**
Avg. filled, maxJobs	100% 16	**4-100%** 256	-(?) 1120	**100%** 570=90*3+50*6
Avg. waiting	**1000% +**	0-150%	-(?)	-(?)
Fin _24h	1160	400	630	3900
Run _24h	12	4	44	344
Act _24h	350	3	4	700
Fail _24h	**150**	0	0	13

PanDA at NRC KI: Текущее состояние (2)



PanDA at HPC KI
задачи, требующие решения

I. Дать пользователям очереди ANALY возможность запускать задачи multicore(MC)/HighMem(HM). («балансировка узлов»)

II. Сбалансировать использование выделенных ресурсов суперкомпьютера. («балансировка очередей»)

PanDA at HPC KI
возможные способы решения (1)

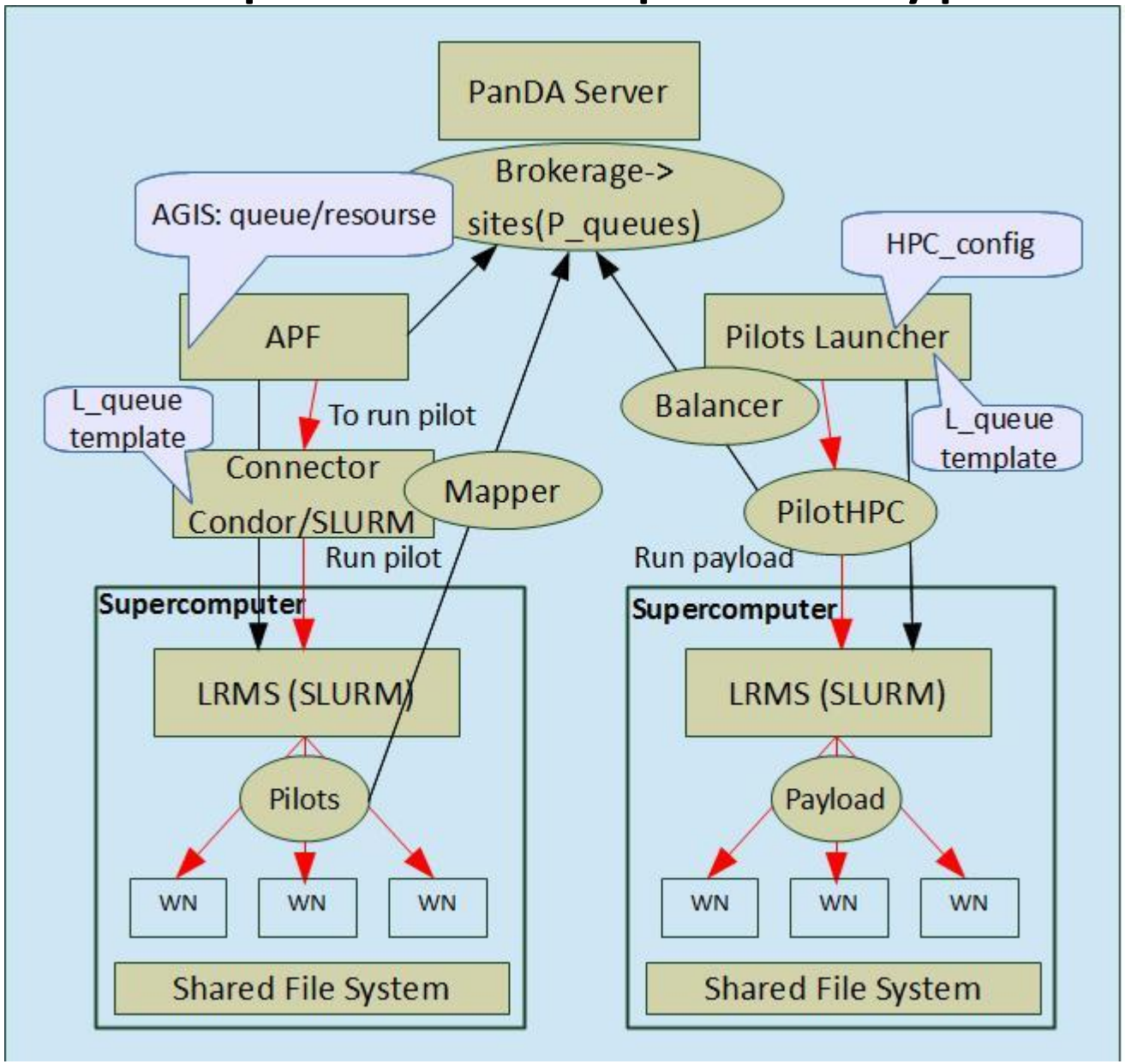
I.1 APF + Condor2SLURM_connector (2 HPC очереди):

поддерживают однородные задачи, а для MC/HM потребуется специальный балансировщик, который будет самостоятельно фиксировать маппинг задач на узлы очередей SLURM.

I.2. Pilot4HPC_KI (аналог ветки Titan (backfill)):

поддерживает MC & HM, но не привязан к системе APF (и 2-м очередям HPC) . Требует доработки для гибкой поддержки MC & HM, в т.ч. поддержки SLURM allocations для произвольных требований ресурсов, которые определены только после получения задачи (от PanDA server at CERN).

Сравнение архитектур



PanDA at HPC KI
возможные способы решения (2)

I.3. Опыт pilot4HPC_KI для задач BIO – локальный PanDA server + однородная очередь (1 ИЛИ 8 ядер на задачу).

Варианты:

1) брать задачу (любую, всегда) и перезапускать её при отсутствии свободных ресурсов (~=метод backfill at Titan);
2) добавить метод (в пилот/сервер), узнающий требования следующей готовой к счёту задачи (т.н. «резервирование задачи»).

PanDA at HPC KI
возможные способы решения (3)

II. Балансирование ресурсов между 2-мя HPC очередями.

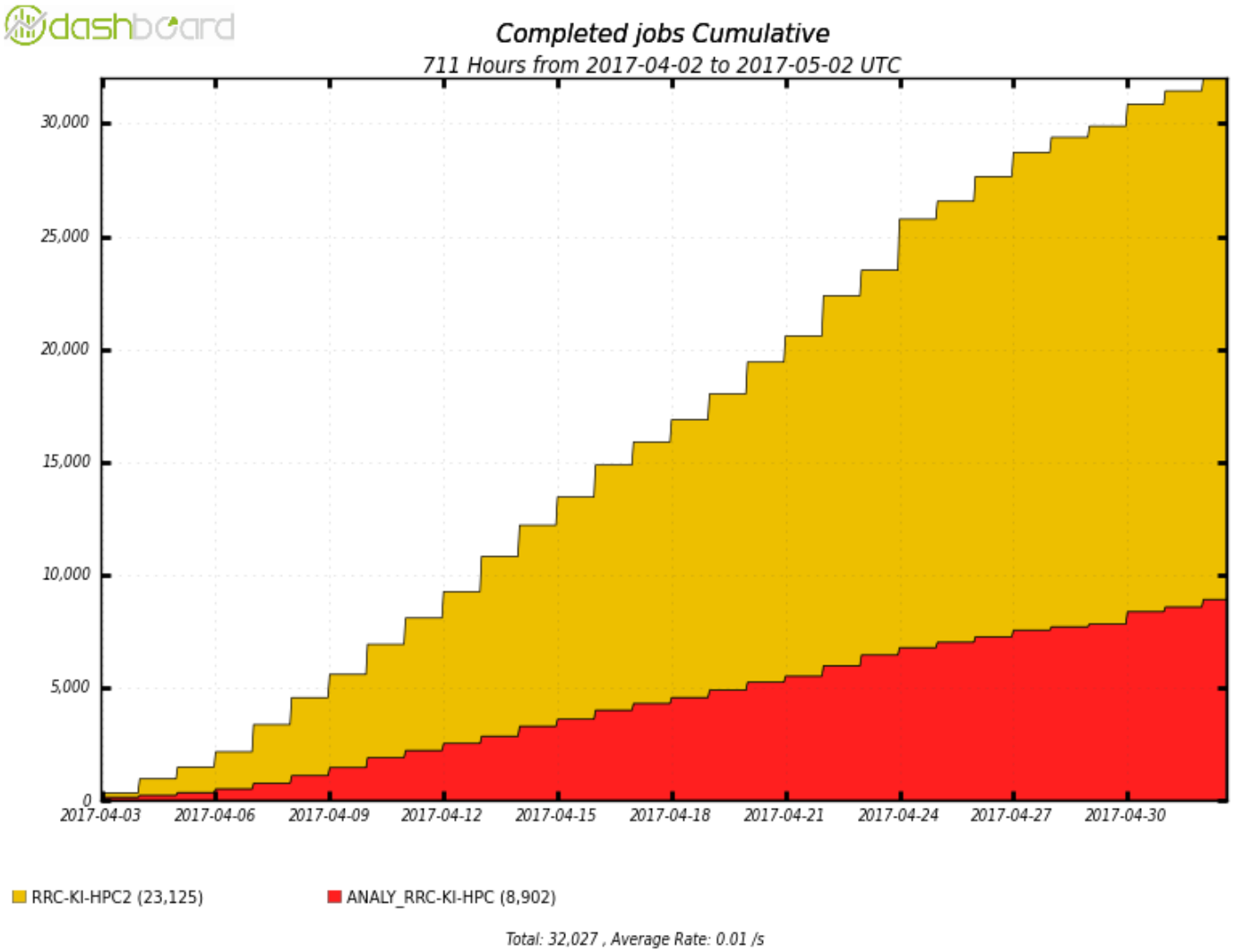
Дано:

PanDA очереди Q1 и Q2, с параметрами: NW_jobs количество ожидающих задач ‘ready’(+’activated’?), количество свободных узлов NFr_nodes.

Алгоритм (простейший):

Если Q1_NFr_nodes >0 и Q1_NW_jobs небольшое (=?), и если Q2_NW_jobs > Q2_MaxJobs, то запускать пилоты для Q2 на ресурсах Q1 (HPC).

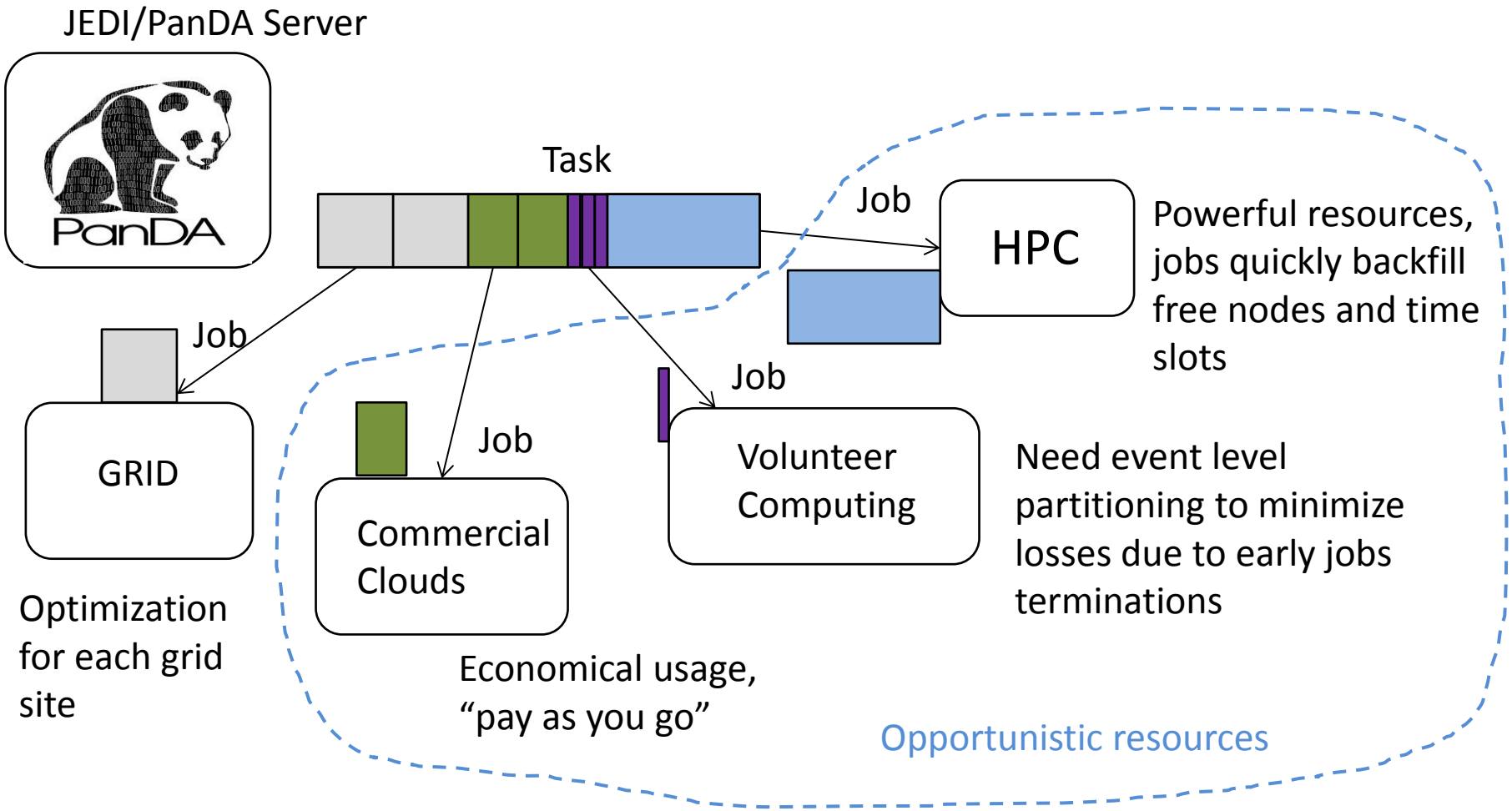
Влияние балансирования ресурсов между 2-мя HPC очередями



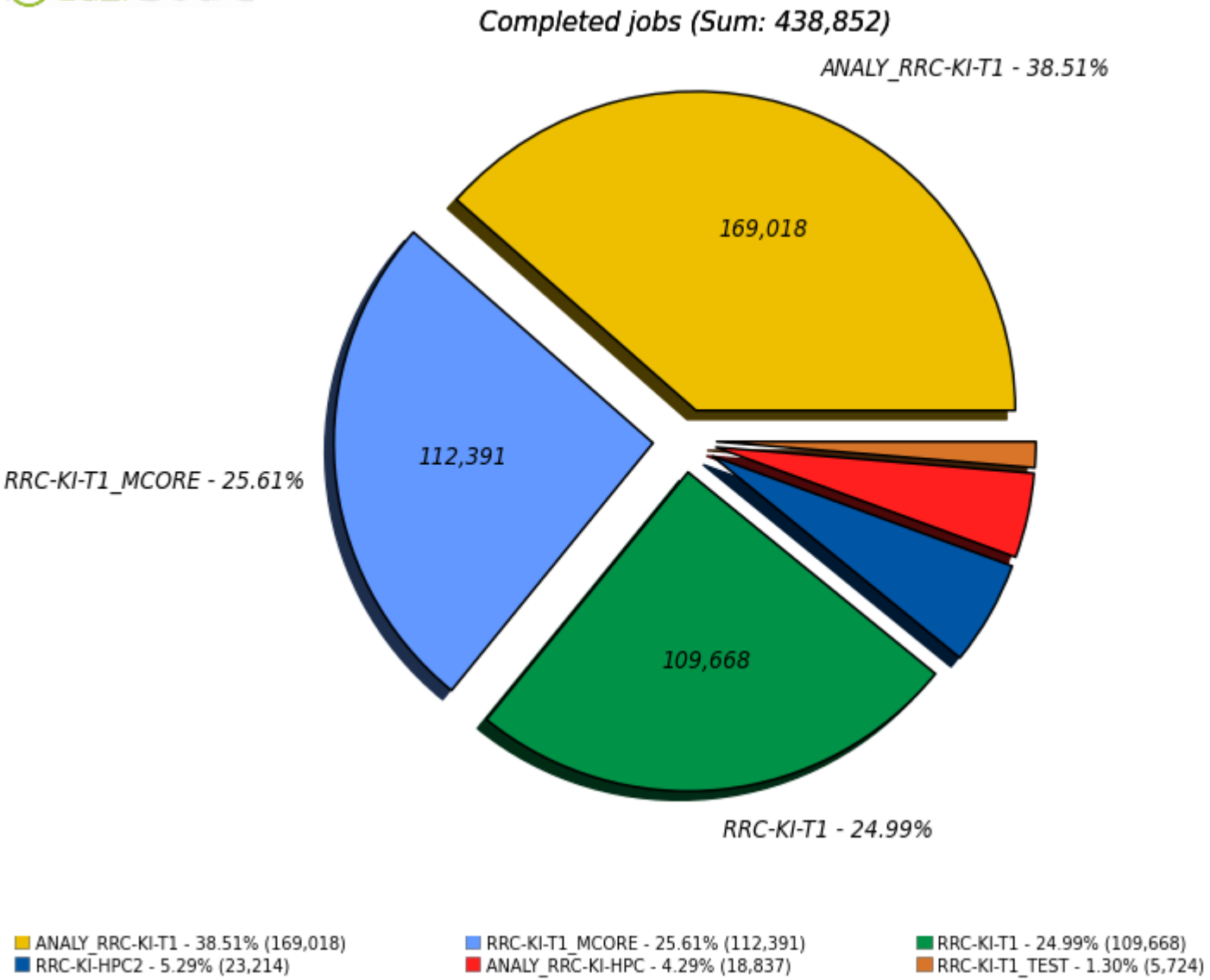
Спасибо за внимание!

Обсуждение?

Dynamic job definition and workload partitioning in PanDA or multiHPC nonHEP projects



dashboard


Completed jobs (Sum: 438,852)
ANALY_RRC-KI-T1 - 38.51%
169,018
RRC-KI-T1_MCORE - 25.61%
112,391
109,668
RRC-KI-T1 - 24.99%
ANALY_RRC-KI-T1 - 38.51% (169,018)
RRC-KI-T1_MCORE - 25.61% (112,391)
RRC-KI-T1 - 24.99% (109,668)
RRC-KI-HPC2 - 5.29% (23,214)
ANALY_RRC-KI-HPC - 4.29% (18,837)
RRC-KI-T1_TEST - 1.30% (5,724)