Evolution de la gestion du système de batch UGE

Réunion des expériences

30 / 01 / 2018

Frédéric AZEVEDO, Nadia LAJILI

CCIN2P3

cnrs

Pourquoi ces évolutions ?

- Configuration devenue complexe avec le temps
  - Nombreux plafonds/limites (complexes / rqs / objectif)
  - Nécessité d'adapter la configuration en fonction du profil des jobs en queue
  - Mécanismes biaisant l'ordonnanceur de batch

- Demandes régulières concernant
  - Niveau de jobs qui semble anormal / "injuste"
  - Productions qui ne s'écoulent pas assez rapidement

Quels objectifs ?

- Pour nous
  - Optimiser le taux de remplissage de la ferme de calcul
  - Faciliter l'atteinte des objectifs des expériences
  - Minimiser les interventions manuelles

- Pour les expériences et utilisateurs
  - Améliorer la gestion des productions par pic
  - Réduire les sources de limitations/plafonds

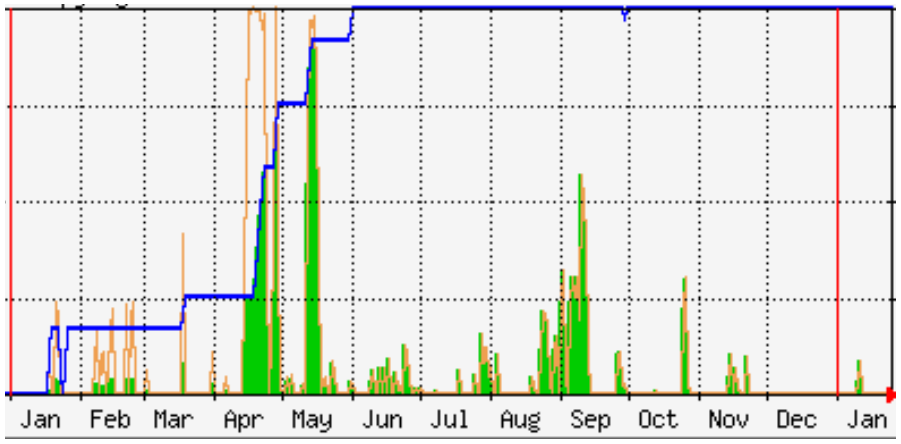
Evolutions

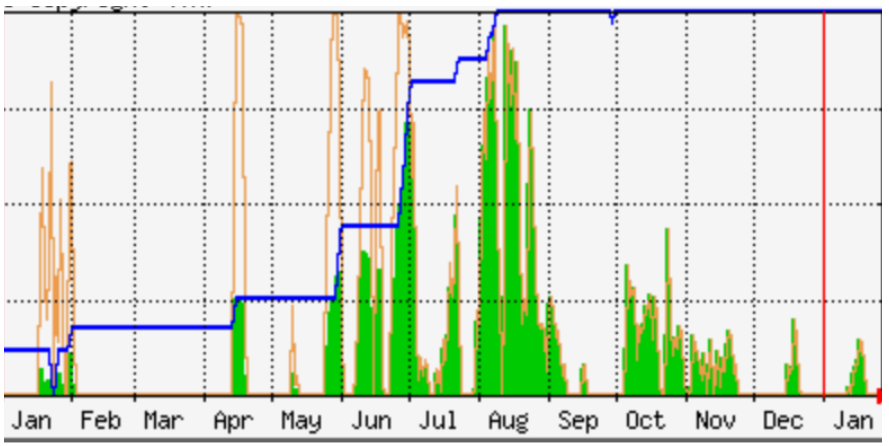
- Optimiser l'utilisation de la ferme en cas de "places libres"
  - Souvent suite à des "grosses" expériences qui ne calculent plus
  - Ne pas privilégier intentionnellement les VOs LHC
  - Laisser UGE assurer le fair share

- "Banaliser" les noeuds de calcul
  - Plus de distinction entre séquentiel / multicore / multiseq
  - Tout est devenu multiseq
    - Un noeud de calcul accepte des jobs mono et multi-coeurs
    - Apporte de la souplesse dans l'optimisation de l'utilisation de la ferme

Evolutions

- Réduire les facteurs limitants
  - Limitation uniquement pour raison de charge / capacité / dysfonctionnement
  - Augmentation systématique des RQS atteints
    - valable pour les RQS stockage (dcache, sps, …)





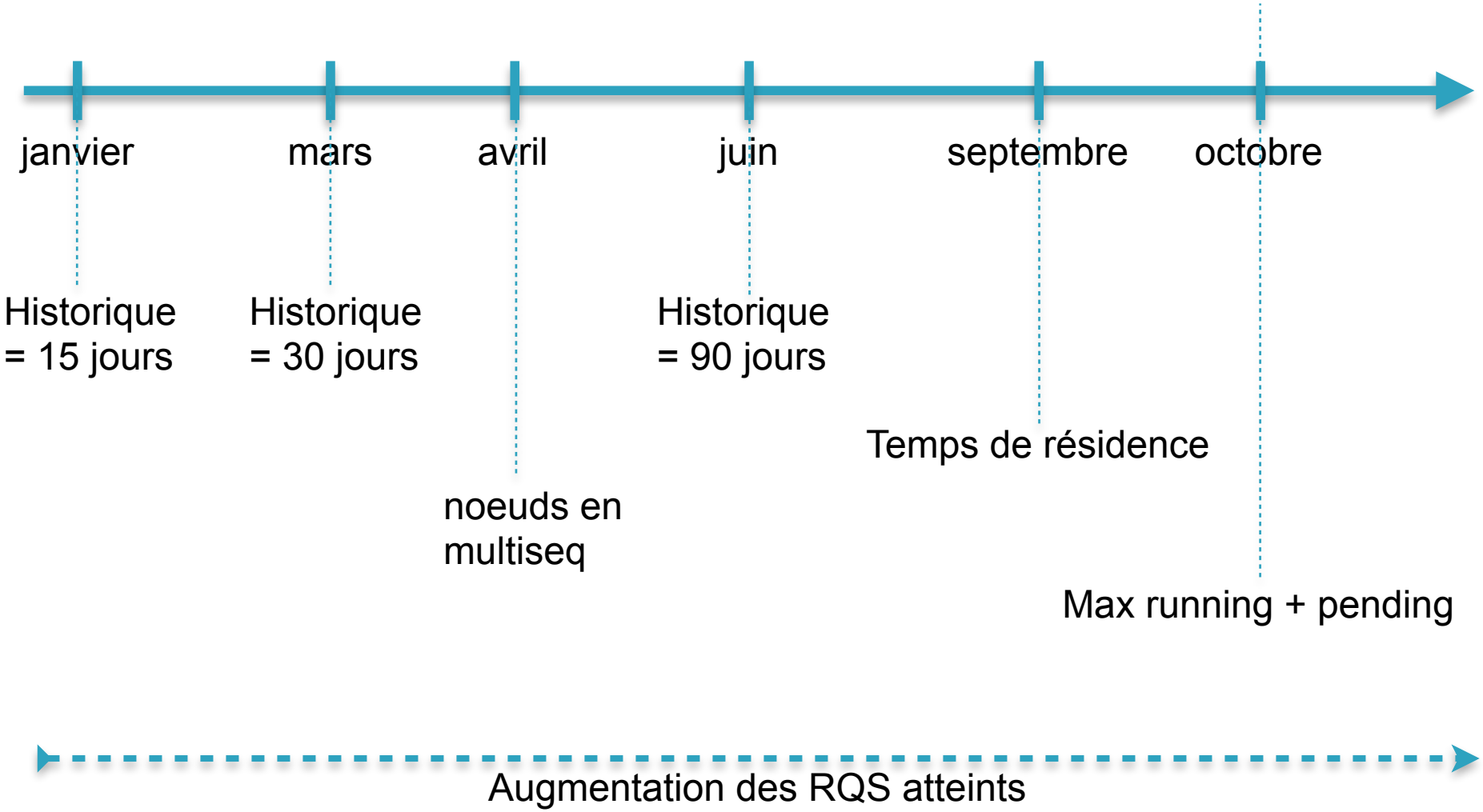
    - valable aussi pour les limites slots

- Augmentation du nombre max de jobs par utilisateur
  - concerne l'ensemble running + pending

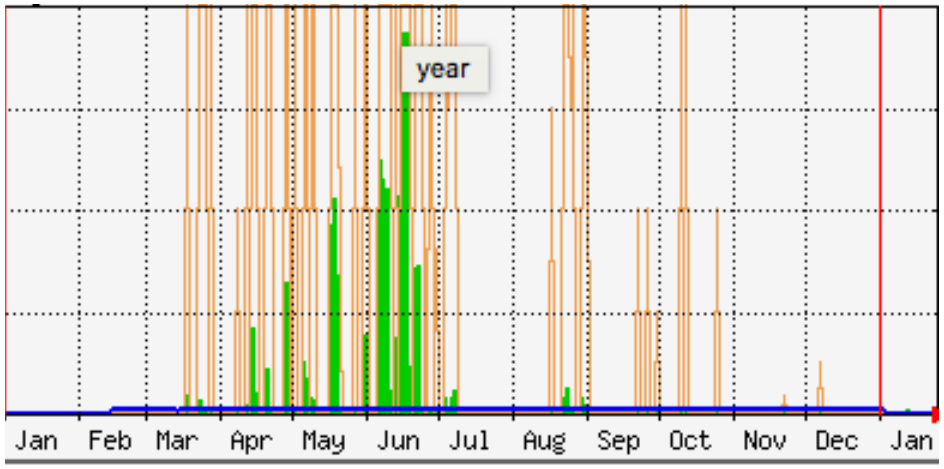
Faciliter les productions par pic

- Modifications dans la configuration du système
  - **Historique** pris en compte pour le **fair share**
    - Progressivement augmenté
      - 24h à 15 jours, puis 30 jours et enfin 90 jours
  - **Temps de résidence** au lieu du cpu consommé pour le **fair share**
    - Les jobs non efficaces ne sont plus favorisés
      - Efficacité cpu : cpu consommé / temps de résidence
- Si cela ne suffit pas : demande de “boost” **justifiée**
  - Cela permet d’augmenter la priorité / objectif d’un groupe
    - Pour une durée définie
    - Au détriment des autres expériences

Récapitulatif

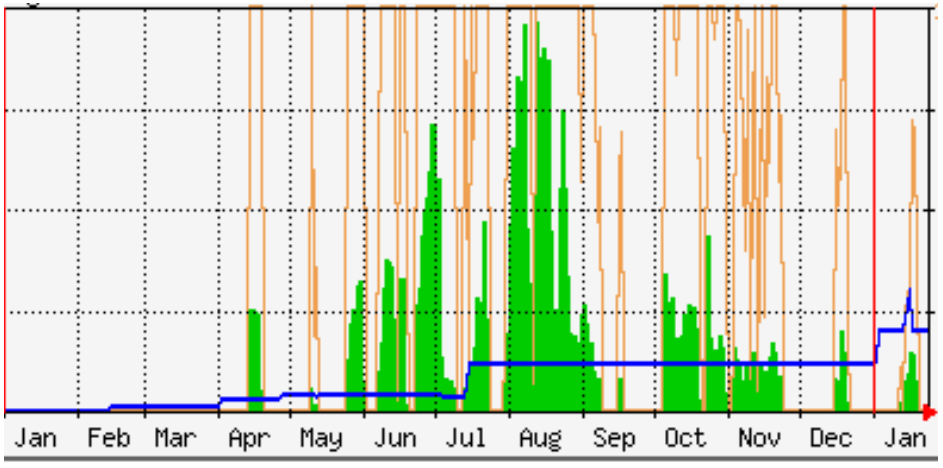


Exemples de productions par pic

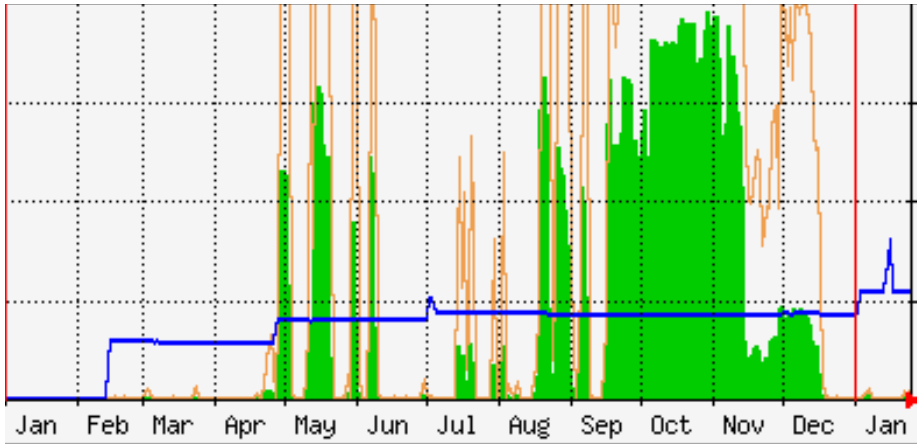


- Objectif : env. 10 slots
- Max atteint : 450 slots

- Objectif : env. 40 slots
- Max atteint : 800 slots

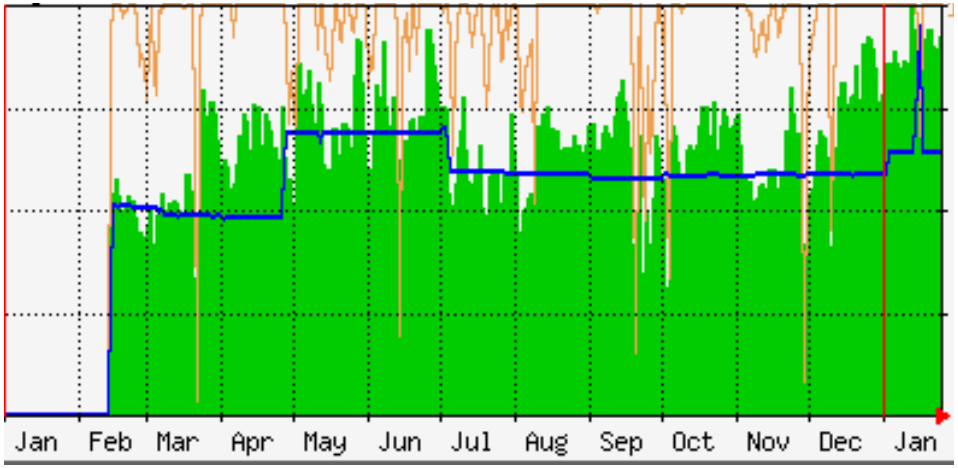


Autres exemples



- Boost de 1 mois
- Objectif : env. 1000 slots
- Max atteint : 5 000 slots

- Production continue
- Fluctuations autour de la valeur correspondant à l'objectif annuel



Retour expériences / utilisateurs

- Avez-vous constaté des changements en 2017 ?
  - Positifs / négatifs
- L'efficacité cpu des jobs
  - Est-il utile que nous vous fassions remonter une efficacité cpu basse/anormale ?
  - Avez-vous ou aurez-vous des jobs avec un profil de plus en plus IO intensif ?
- Le critère du temps d'attente en queue
  - Est-il important pour vous ?
  - Pourquoi ? Pour tester/valider ?
- Y a-t-il des besoins non satisfaits coté calcul ?

Rappels

Arbitrage des ressources

- Pas de requête cpu, pas de jobs (comme en 2017)

- Au vue des contraintes budgétaires et administratives
  - Rarement possible d'acheter avant février
  - Matériel disponible plutôt vers avril
  - La demande peut être difficile à satisfaire au premier trimestre

- Il n'existe pas de limite en place sur l'atteinte de l'objectif annuel (en HS06)
  - Réflexion sur une possible re-évaluation des priorités des groupes ayant atteint leur objectif annuel
    - Bien évaluer sa demande de cpu

Bonnes pratiques

- Exprimer explicitement les ressources nécessaires à la soumission
  - Mémoire, cpu, disque, rqs, plateforme
  - Le projet batch (ex : P_mongroupe)
  - Laisser UGE choisir la queue adaptée au job
- De préférence, valider les jobs par un/des jobs interactifs
- Assurer un flux constant de jobs (si possible)
- Eviter les accès intensifs aux systèmes de stockage
  - Privilégier une copie locale dans $TMPDIR
- Documentation utile
  - Wiki utilisateur : utiliser le système de batch
  - FAQ Calcul sur OTRS

Merci

-

Questions ?