

# Os Filtros de Alto Nível do Experimento ATLAS

José Manoel de Seixas e André Rabello dos Anjos  
[seixas@lps.ufrj.br](mailto:seixas@lps.ufrj.br), [Andre.dos.Anjos@cern.ch](mailto:Andre.dos.Anjos@cern.ch)

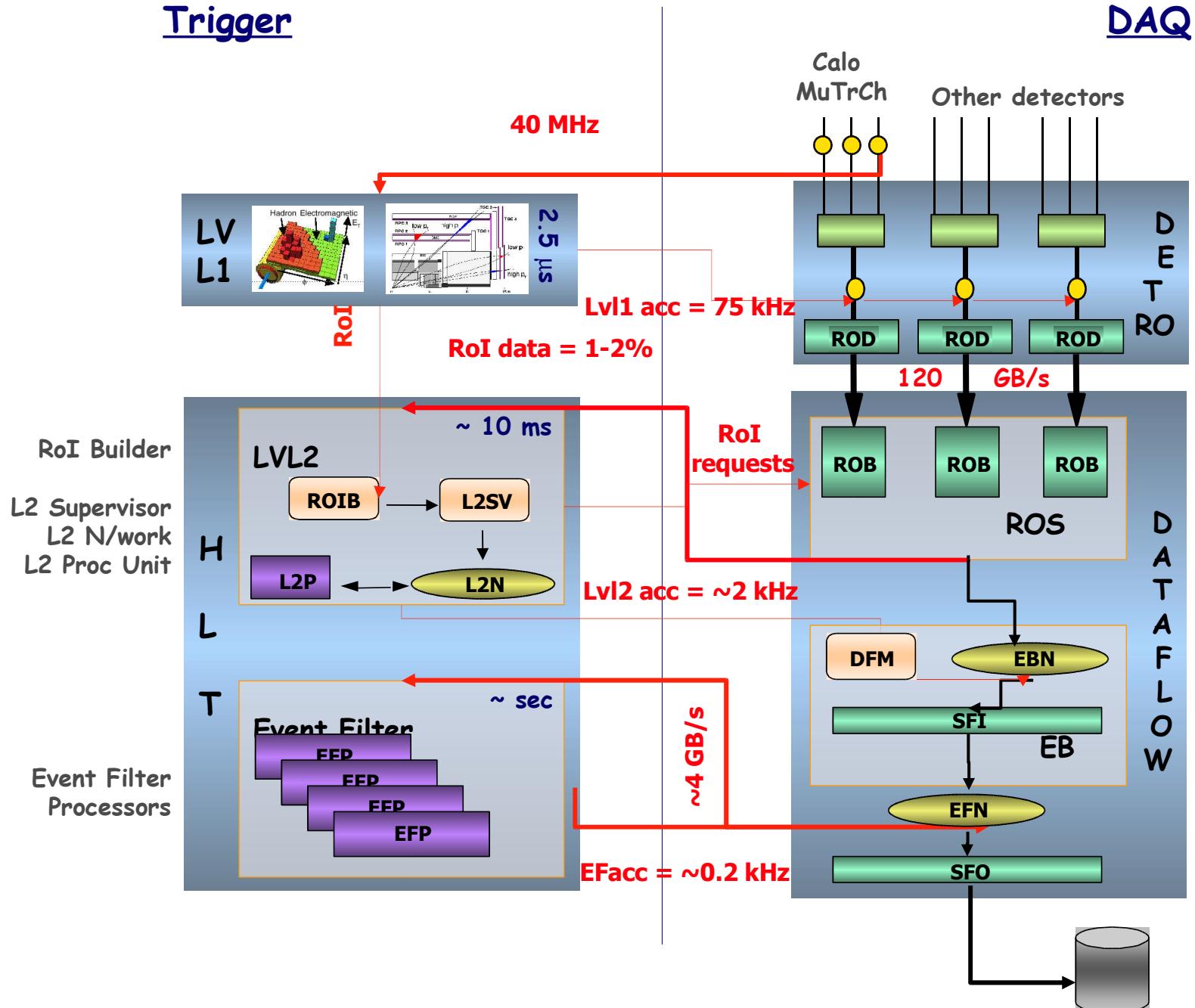
UFRJ/COPPE/LPS – 24 de agosto de 2004

# Conteúdo

- O Sistema de Filtragem do Experimento ATLAS
- O LVL2
- O Filtro de Eventos ou LVL3
- Athena
- Interoperação Athena - HLT
- Conclusões

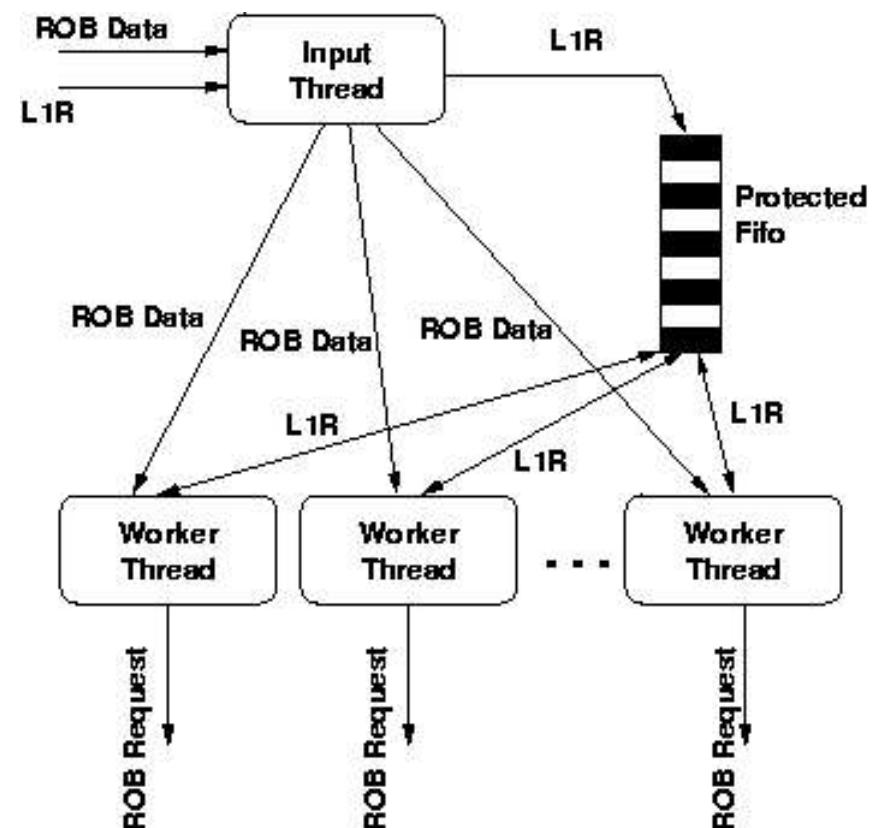
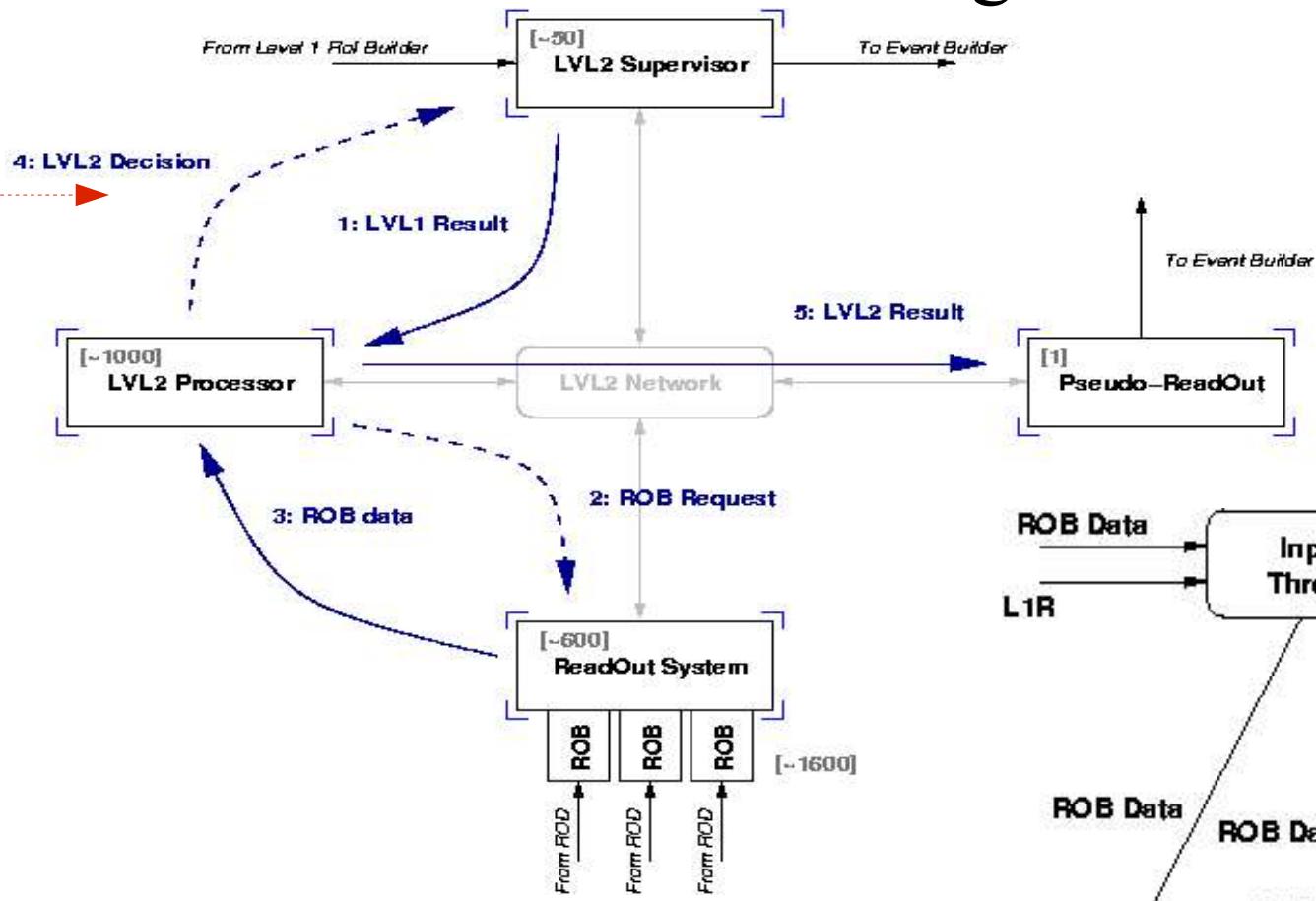
# Sistema de Filtragem do ATLAS

## Trigger



# O Segundo Nível de Filtragem

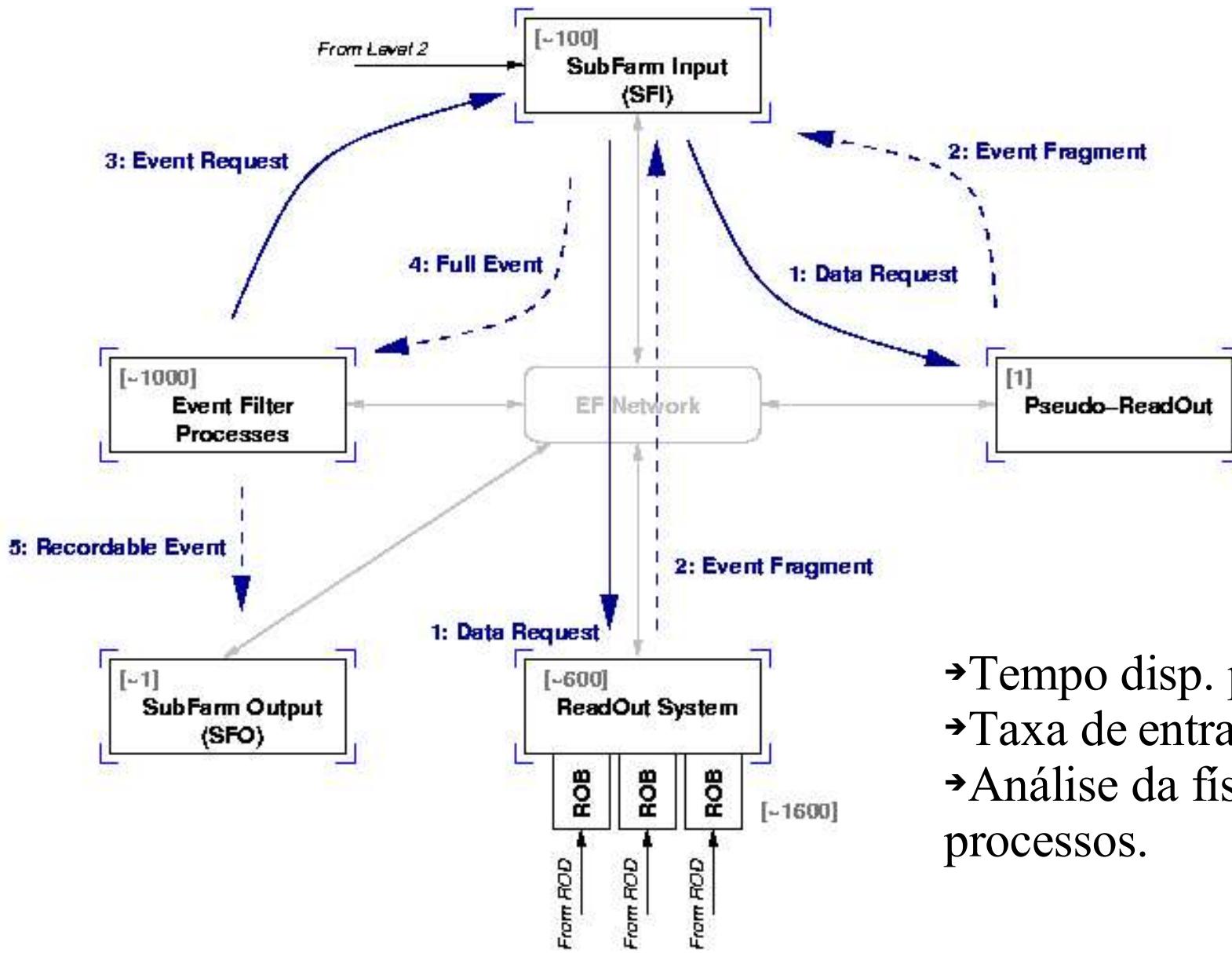
## - Visão do Evento: Região de Interesse -



- Para 100 kHz de entrada e 1.000 processadores, o tempo médio de execução por evento é de **10 ms**.
- A análise física é feita em tarefas.

# O Filtro de Eventos (Nível 3)

## - Visão do Evento: Evento Completo -



- Tempo disp. por evento: 1 s
- Taxa de entrada: 1 kHz
- Análise da física feita em processos.

# Processamento *Offline*: Athena

- O Athena é um conjunto de bibliotecas que implementa: análise física, reconstrução de eventos, simulação e, finalmente, filtragem *online* de eventos, de forma específica ao experimento ATLAS;
- Baseado na arquitetura Gaudi para o processamento de física (algoritmos, ferramentas e serviços bem separados dos dados);
- Totalmente escrito em C++, usando seu potencial mais amplo;
- Contém contribuições dos Sistemas de Detecção e do Sistema de Filtragem (Seletor de Regiões, *Steering*, Algoritmos de Filtragem);
- É um projeto bastante grande, unindo centenas de programadores espalhados pelo mundo.

# Vantagens de unir *online* e *offline*

- Testes de eficiência e depuração podem ser executados *offline*. Durante a tomada de dados, há somente verificação dos resultados;
- “Sintonia fina” de cortes e eficiências migrando algoritmos de filtragem entre o segundo e terceiro níveis de filtragem;
- Existirá apenas 1 conjunto de aplicativos que define as capacidades de análise física do ATLAS. O Athena será estudado e melhorado por todas as partes, ao invés de existirem sub-projetos isolados.

# Problemas e Incompatibilidades

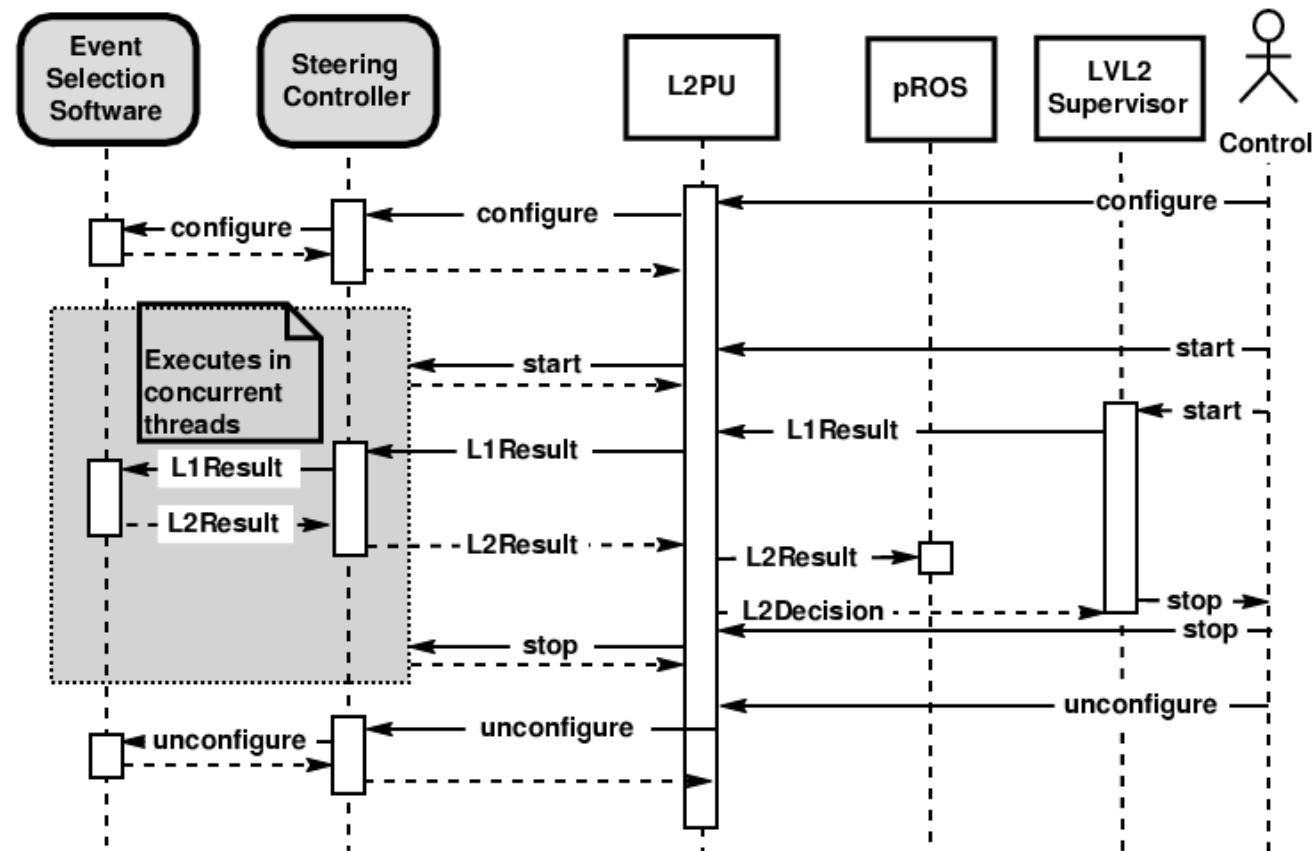
- O Athena foi concebido para ser executado em um único processo (não multi-tarefa) e iterativamente na maior parte dos casos.
  - O sistema *online* é multi-tarefas: múltiplos eventos podem ter que ser executados concurrentemente
  - O sistema de configuração do software do Sistema de Filtragem é centralizado em bancos de dados
  - O sistema de *log* de erros é centralizado e subjugado ao operador de um *run*.

Da mesma forma são os dados de monitoração:  
contagens, histogramas, etc.

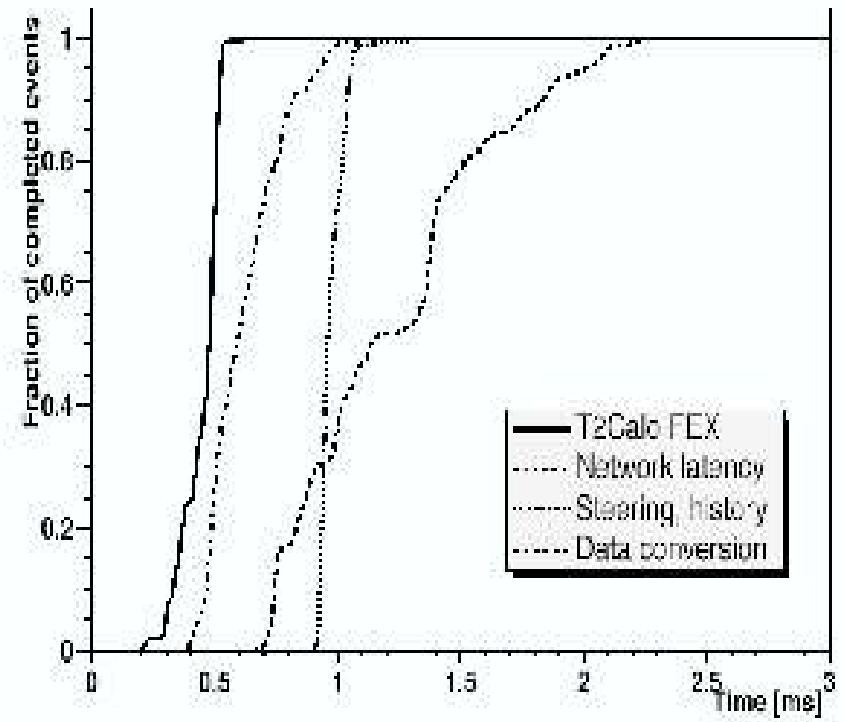
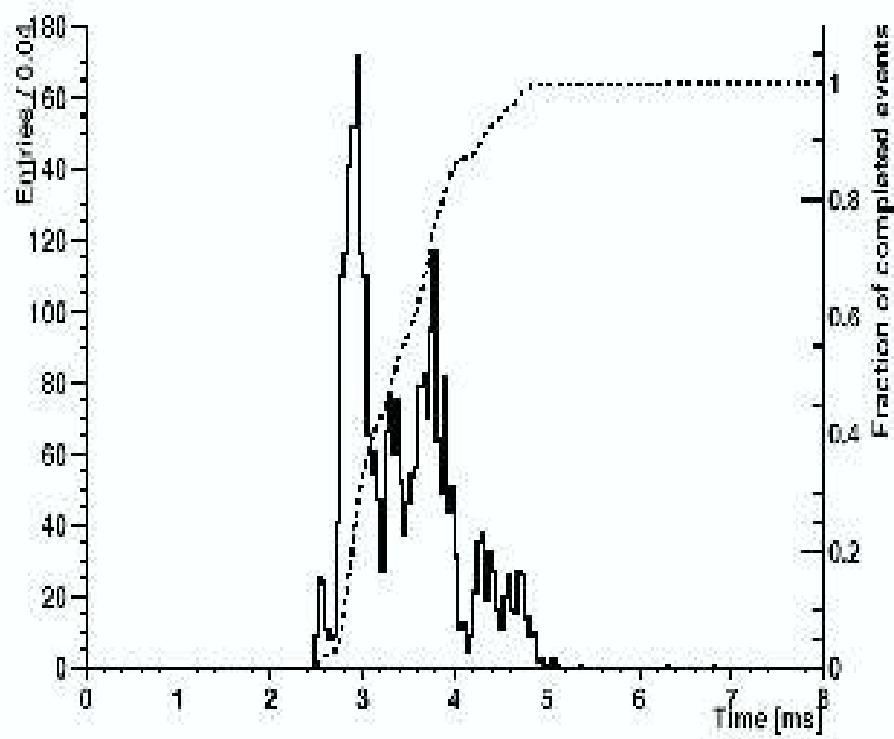
## Como resolver estes problemas?

# Solução: PESA Steering Controller

- Este componente cria um *wrapper* ao redor do Athena, “simulando” um ambiente de trabalho compatível com o ambiente *offline*.
- No exemplo abaixo, encontra-se a interação entre a L2PU (LVL2) e este componente. Os comandos de controle do sistema *online* são repassados e interpretados pelo PSC.
- Os arquivos de configuração são encontrados também pelo PSC e repassados ao Athena.
- Serviços globais são duplicados para garantir a execução em múltiplas tarefas de processamento
- Mensagens de erro e depuração são redirecionadas para os devidos serviços de monitoração e controle de erros *online*.
- O *overhead* é desprezível.



# Exemplo: execução da filtragem de múons no LVL2



Neste exemplo, é possível observar que o tempo de processamento dedicado ao PSC (indicado por “Steering history”), é diminuto se comparado com o tempo gasto, e.g., na conversão de dados (“Data conversion”). Resultados similares foram obtidos para outras funções.

# Conclusões

- Utilizando-se da arquitetura Gaudi/Athena disponível no ATLAS, é possível implementar uma interface de “cola” entre dois sub-sistemas com prioridades bastante distintas: *offline* e *online*.
- Este componente provê, para o ambiente *offline*, a sensação de estar sendo executado como de hábito, e ao ambiente *online*, uma interface padronizada.
- O tempo de execução extra é desprezível
- Com esta solução, o desenvolvimento de algoritmos de filtragem pode ser feito separadamente do desenvolvimento do sistema *online*. Isto inclui depuração e testes de eficiência.
- Uma vez que tanto no LVL2 quanto no EF, o mesmo ambiente de Filtragem está disponível, é possível controlar a eficiência dos cortes e migrar algoritmos de um lado para o outro, otimizando a operação do sistema de filtragem.