Message Aggregation

Danny Auble (SchedMD), Matthieu Hautreux (CEA)
Yiannis Georgiou, Martin Perry (Bull)

Slurm User Group 2015

Message Aggregation Overview

- Essentially the reverse of the message forwarding/fanout mechanism used to reduce the load on the controller for broadcast messages
  - Messages coming from the slurmctld have always worked in a tree fanout along with the responses
  - Messages originating from the slurmd have always gone directly to the slurmctld outside of any tree
- Designed to improve communication originating from outside the slurmctld
  - reduce overhead on the slurmctld
  - reduce number of incoming TCP connections to serve
  - Able to handle many messages inside a single lock instead of fighting for locks from different messages
- May be enhanced in the future to support additional message types and destination nodes (not just the controller)

Message Aggregation Design

- New Message Types
  - MESSAGE_COMPOSITE
  - RESPONSE_MESSAGE_COMPOSITE
- Supported Messages
  - Epilog Complete
  - Node Registration
  - Complete Batch Script
  - Step Complete

Message Aggregation Design

Leaf Nodes, Message Collector Nodes & Destination Nodes

- Leaf Node
  - A node that originates an epilog complete message, or other message type that is eligible for Message Aggregation.
- Message Collector Node
  - A node that receives and collects messages in a Messages Collection, and originates and sends a composite message built from a Messages Collection.
- Destination Node
  - A node that is the final destination of a composite message.
  - Each node involved in Message Aggregation may be a collector node only, both a leaf node and collector node, or a destination node. The only destination node currently supported is the node running slurmctld.

Message Aggregation Design

Messages Collection & Messages Collection Window

- Messages Collection
  - A collection of messages on a message collector node with the same destination.
- Message Collection Window
  - The period during which a single messages collection is built
  - Defined by a maximum number of messages in a collection and a maximum elapsed time
  - Started when the first message in a messages collection is collected
  - Expires when either the maximum number of messages is reached or the maximum elapsed time is reached, whichever occurs first

Message Aggregation Design

Routing

- The route used by message aggregation to send a message from its originating node, through a series of one or more message collector nodes, to its destination node is provided by the route plugin. The reverse route is used to send a response message from its originating node back to the originating node of its associated composite message.

Changes to Slurm Daemons

- Slurmd
    - When a supported message type originates
        - It is collected in the messages collection for destination slurmctld
        - If a return is expected the slurmd waits until a response is given from the tree and then handles it as if it were talking directly to the slurmctld
        - If no return is expected message is sent and the slurmd is done with the message on success to the next node.
    - When a slurmd is acting as a Collector node
        - It is collected in the messages collection for the next hop in the tree
        - When the message collection window expires, a composite message is built from the messages collected and sent to the next node on the route to the destination node (either a message collector node or the destination node itself).
        - The messages collection for the destination is then reset
    - When a response message is received
        - The slurmd processes any messages for itself
        - Passes the rest on down the tree to the next hop

Changes to Slurm Daemons

- Slurmctld
  - When a composite message is received
  - The individual messages in its messages collection are extracted and processed just as if they were processed separately
  - A simple combined response message is sent back the exact same tree they came up

Configuration

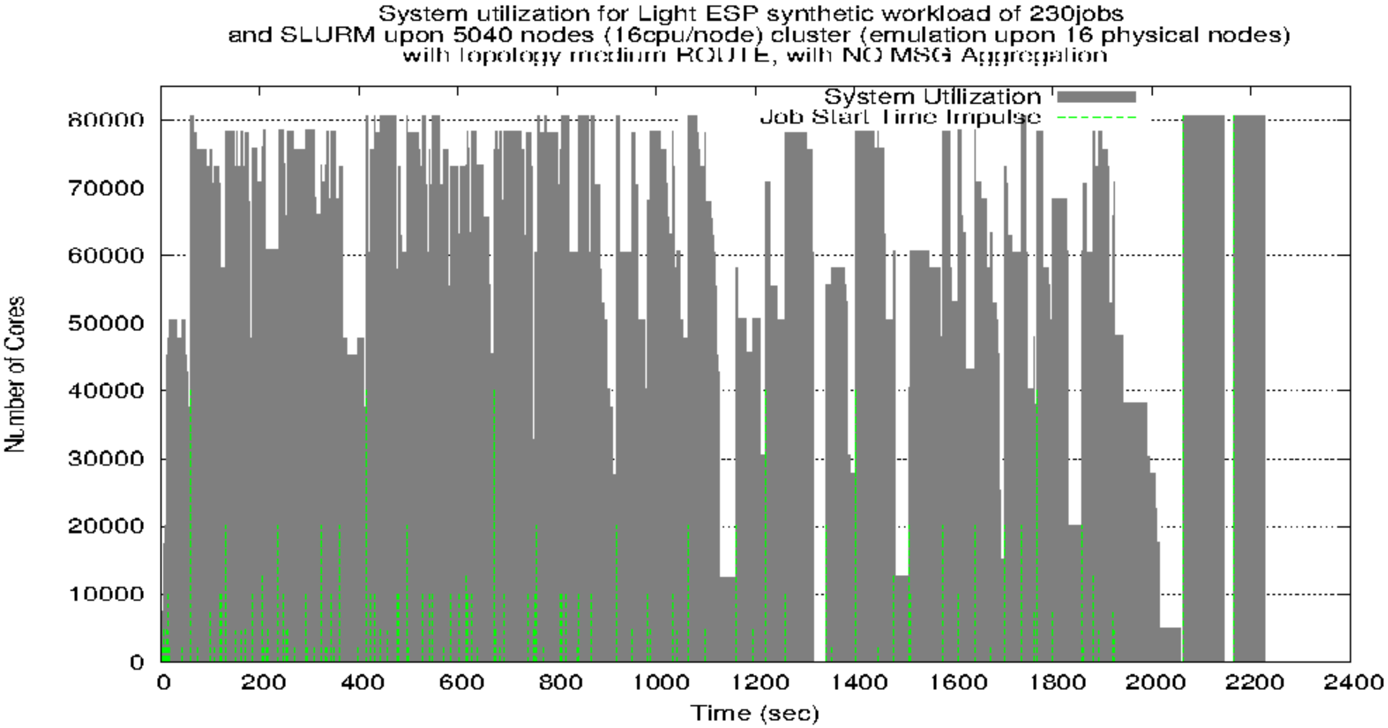
- Message Aggregation is disabled by default
- Enabled with new slurm.conf parameter
  - MsgAggregationParams
    - Defines the message collection window size as a maximum number of messages and maximum time (in milliseconds).
  - Example
    - MsgAggregationParams=WindowMsgs=10,WindowTime=100

Experiments – Testbed

- Motivations for message aggregation optimizations based on the article published in JSSPP-2012 [1]
- Experiments repeated now to validate the new developments
- Consist of executing the Light-ESP synthetic workload composed of 230 jobs of 14 different job profiles (sizes, execution times)
- Deploy 2 different emulated clusters with 5040 and 10080 nodes with 16 cores / node using 18 physical nodes
  - Upon an bullx B510 cluster with Intel Sandybridge (16 cores/node, 64GB)
  - Using “multiple-slurmd” emulation technique
  - Route/topology + defer parameters activated (slurm.conf)
- Comparison between -NO vs WITH- Message Aggregation (for both clusters)
  - System Utilization / Jobs Waiting times
  - Number of messages exchanged throughout the workload

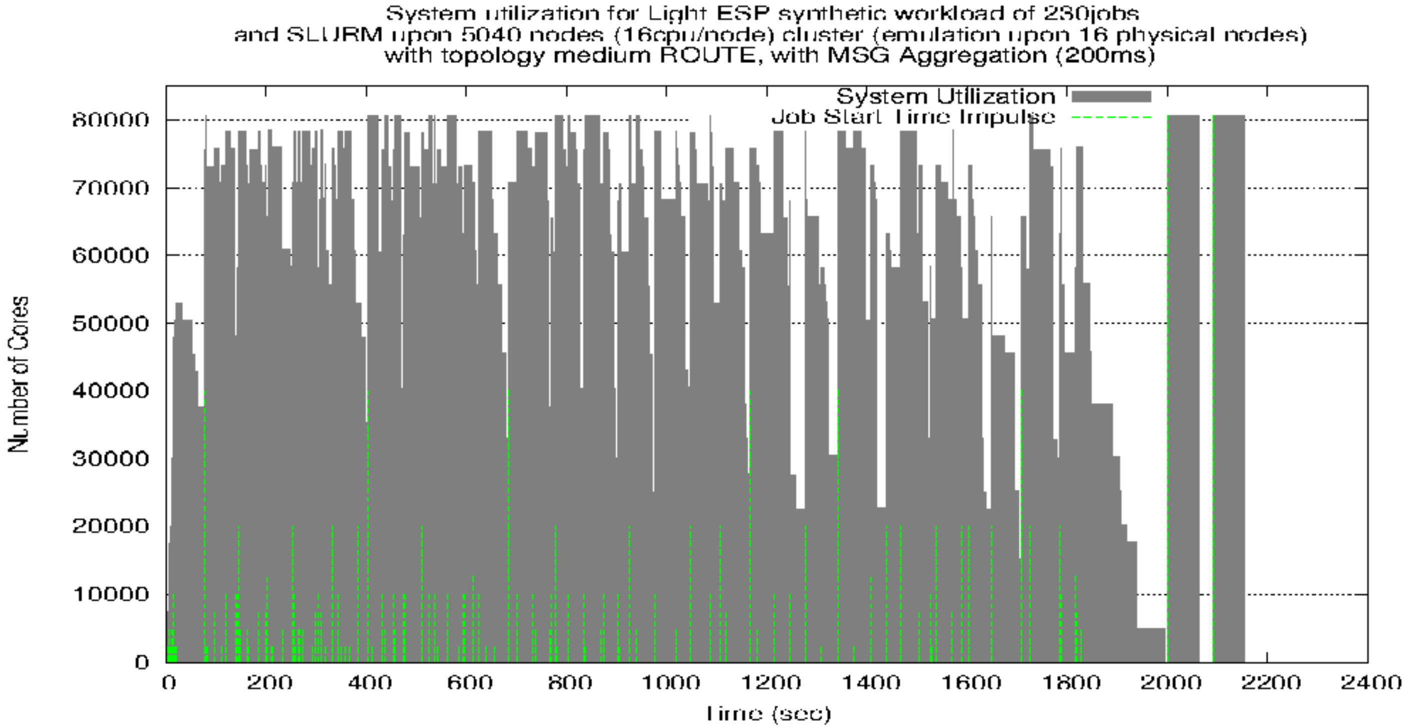
[1] Yiannis Georgiou, Matthieu Hautreux Evaluating Scalability and Efficiency of the Resource and Job Management System on Large HPC Clusters. JSSPP 2012: 134-156

System Utilization 5040 nodes cluster

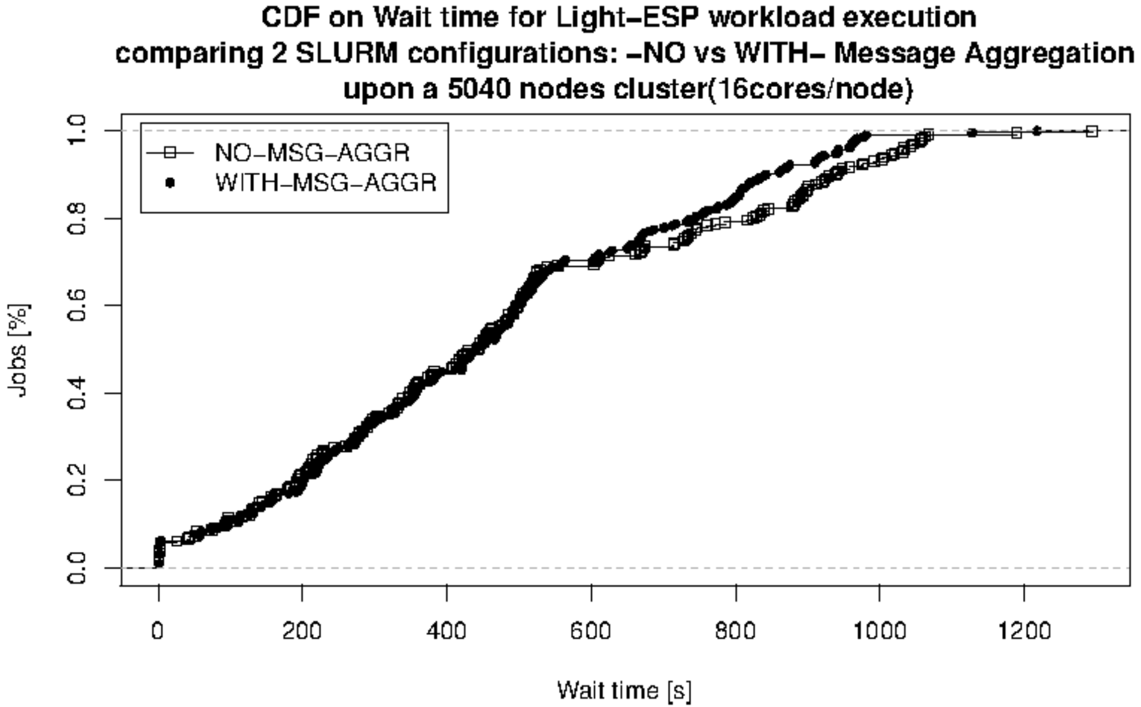


System utilization for Light ESP synthetic workload of 230jobs
and SLURM upon 5040 nodes (16cpu/node) cluster (emulation upon 16 physical nodes)
with topology medium ROUTE, with NO MSG Aggregation

System Utilization 5040 nodes cluster



CDF on Wait time 5040 nodes cluster



sdiag results 5040 nodes cluster

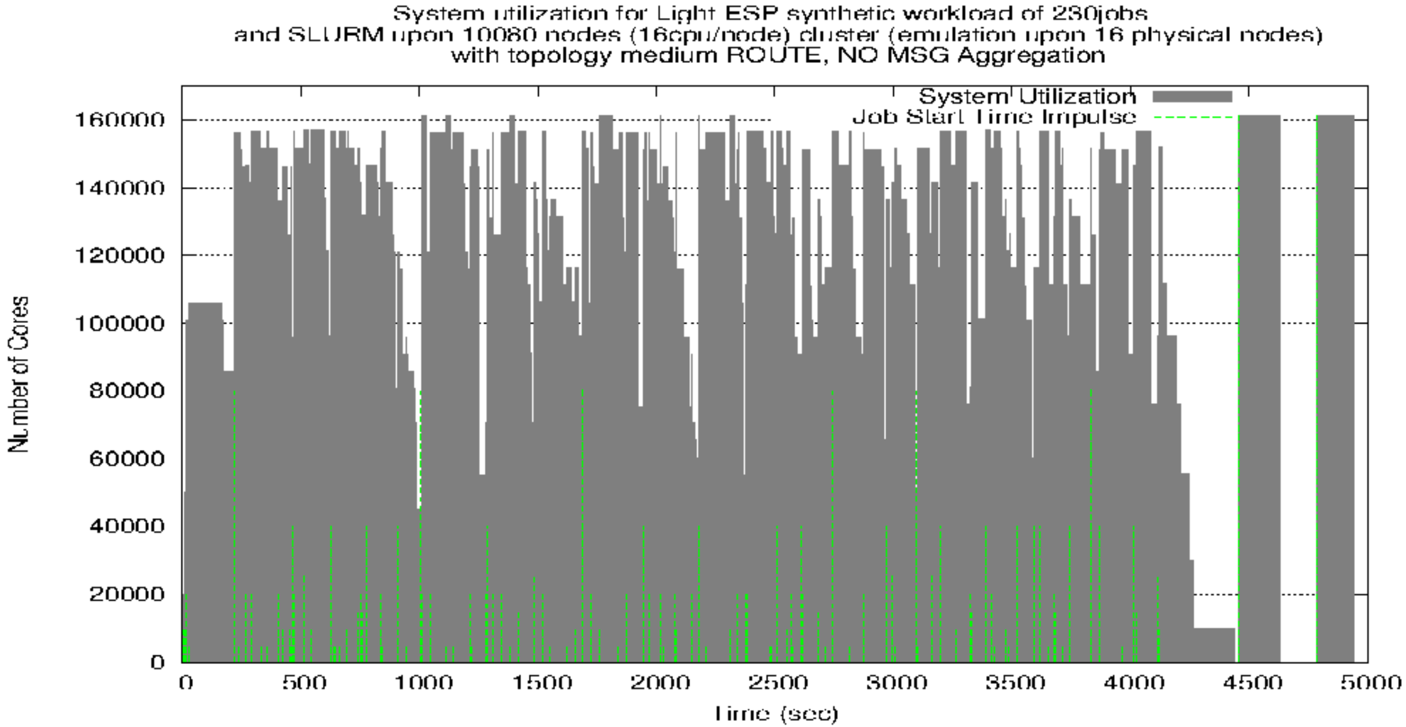
- sdiag result after the end of the workload execution NO-MSG-AGGR

Messages	Count	Average Time (sec)	Total Time (sec)
Epilog-Complete	115324	0.008	1006
Node-Registration	1300	0.122	159

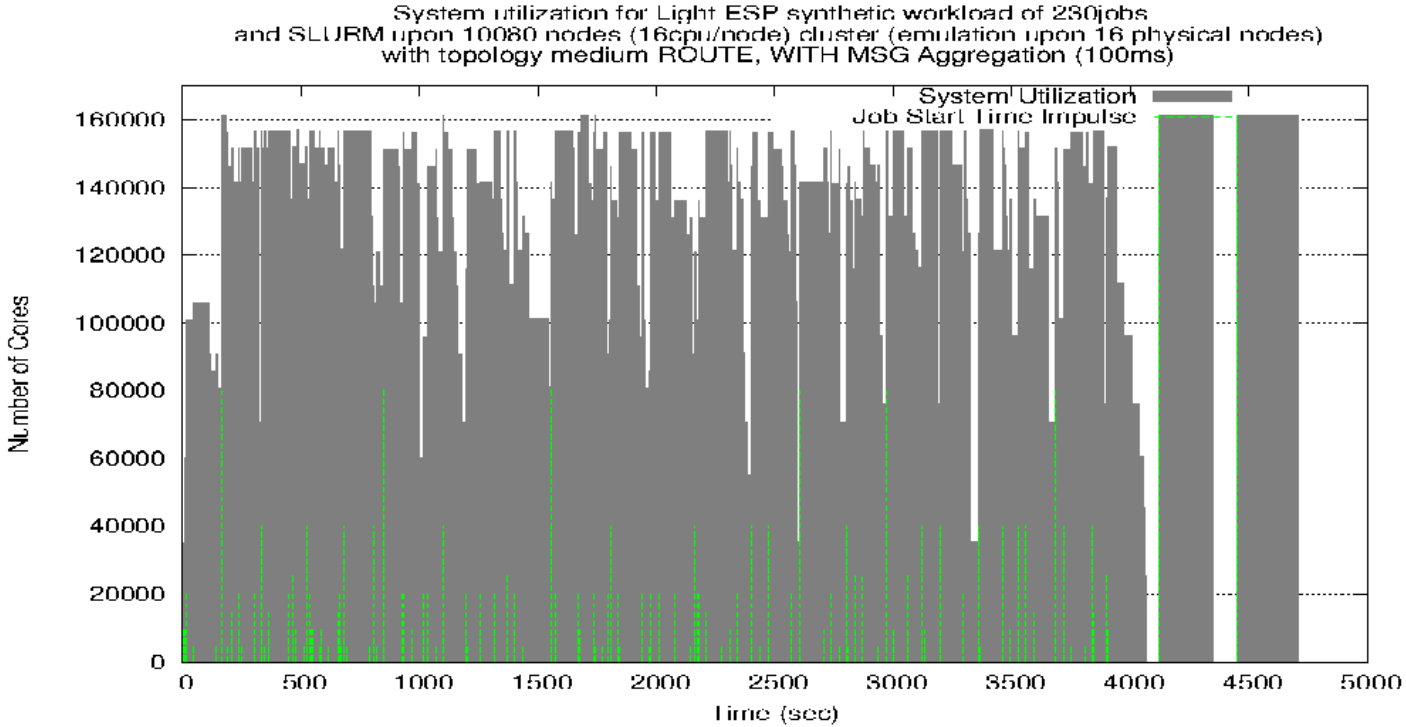
- sdiag result after the end of the workload execution WITH-MSG-AGGR (200sec)

Messages	Count	Average Time (sec)	Total Time (sec)
Composite	2580	0.1	26

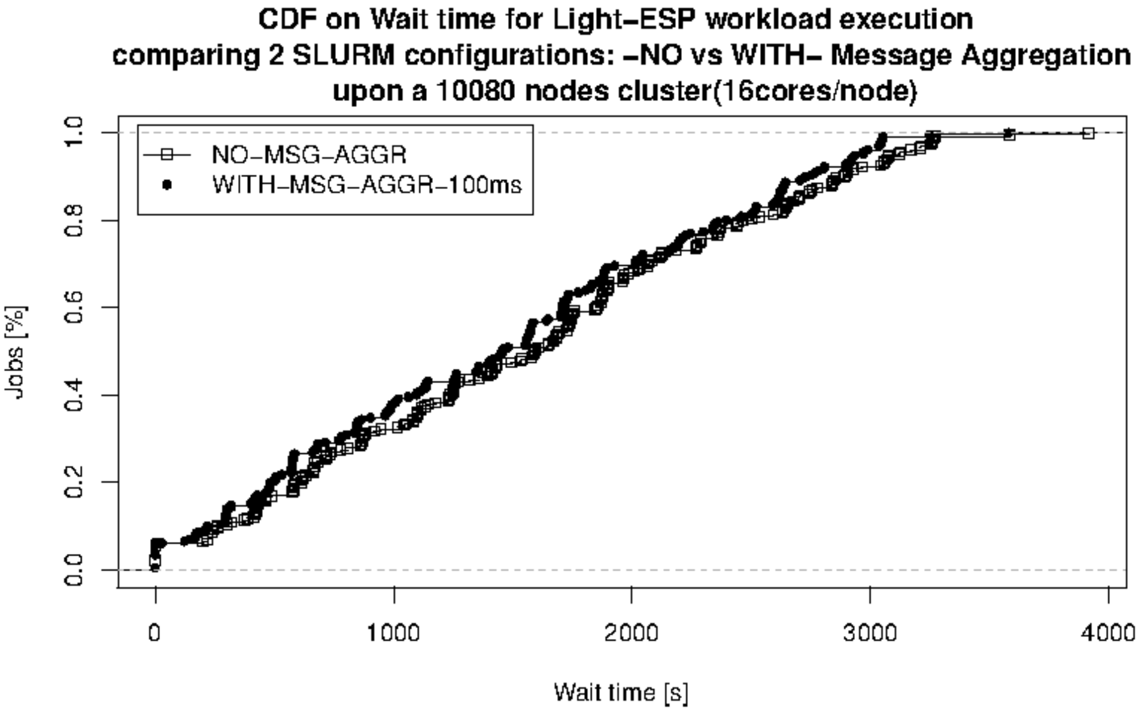
System Utilization 10080 nodes cluster



System Utilization 10080 nodes cluster



CDF on Wait time 10080 nodes cluster



sdiag results 10080 nodes cluster

- sdiag result after the end of the workload execution NO-MSG-AGGR

Messages	Count	Average Time (sec)	Total Time (sec)
Epilog-Complete	230391	0.024	5626
Node-Registration	2550	0.07	194

- sdiag result after the end of the workload execution WITH-MSG-AGGR (200sec)

Messages	Count	Average Time (sec)	Total Time (sec)
Composite	7114	0.046	331