Razorback Requirements Document

**Dispatcher**

**Block Disposition**

**Request Thread**

Version 0.2

Revision 2

5/5/2011

Contents

[Introduction 3](#_Toc292376398)

[Customer Focus Statement 3](#_Toc292376399)

[Requirements 3](#_Toc292376400)

[Messaging 3](#_Toc292376401)

[Locking 3](#_Toc292376402)

[Database 3](#_Toc292376403)

[Implementation 4](#_Toc292376404)

[Metrics 4](#_Toc292376405)

[QA 5](#_Toc292376406)

[Test 1 5](#_Toc292376407)

[Future Work 6](#_Toc292376408)

[Enhancements 6](#_Toc292376409)

6

## Introduction

The Dispatcher Request Thread processes requests for block dispositions from nuggets that are capable of submitting block data.

## Customer Focus Statement

This thread provides the following benefits:

* The nuggets will not transmit blocks that have been seen before.
* The nuggets will not touch the database or the global cache directly.

## Requirements

### Messaging

* Dispatcher must receive requests to a queue designated to it.
* Dispatcher must send responses to a queue dedicate to the nugget that requested the disposition.

### Locking

* Access to the global cache must be locked with a mutex.

### Database

* Must look up the block in the database if there is a miss in the global cache and update the cache appropriately.
* Must generate an event in the database when the request is received.

## Implementation

When a precedence request is received the thread will attempt to resolve the two threat flags fields for a given block from the global cache and the database and return this state back to the nugget that requested it.

The system will first check the global cache for an entry, if one is not found it will check the database, if there is no DB entry it will create one with the pending and can haz flags set.

After this is complete it will create the event record and return the result to the requestor.



## Metrics

Cumulative time for request processing

Maximum time per request processed

Minimum time per request processed

Number of requests processed

## QA

### Test 1

* Submit a block and create a block in the DB
* Submit a block and get the correct result from the DB
* Submit a block and get the correct result from the global cache

## Future Work

### Enhancements

TBD.