FlexiShard

Tirathraj Ramburn & Dhrubajyoti Goswami

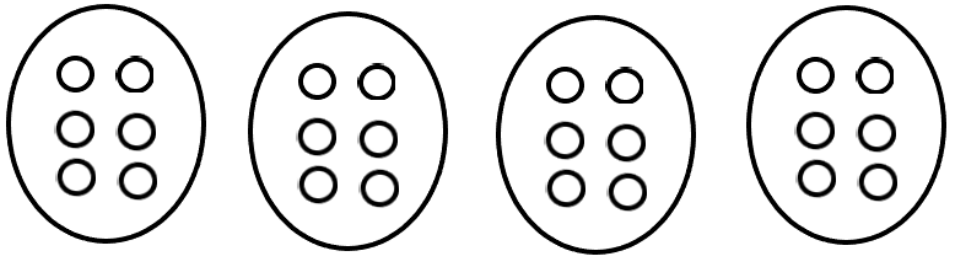
Concordia University, Canada

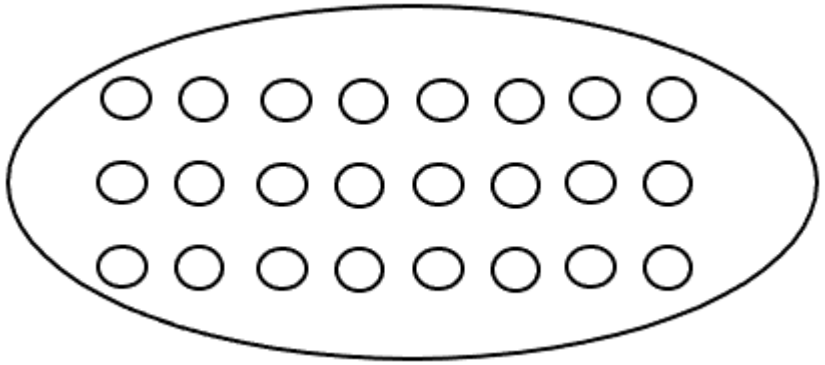
FlexiShard: a Flexible Sharding Scheme for Blockchain based on a Hybrid Fault Model

Agenda

- Bitcoin & Scalability
- Sharding & Shard Size
- Network Model & Intra-Shard BFT Consensus
- Hybrid Fault Model
- Flexible BFT
- FlexiShard & its Schemes

Bitcoin does not Scale Out

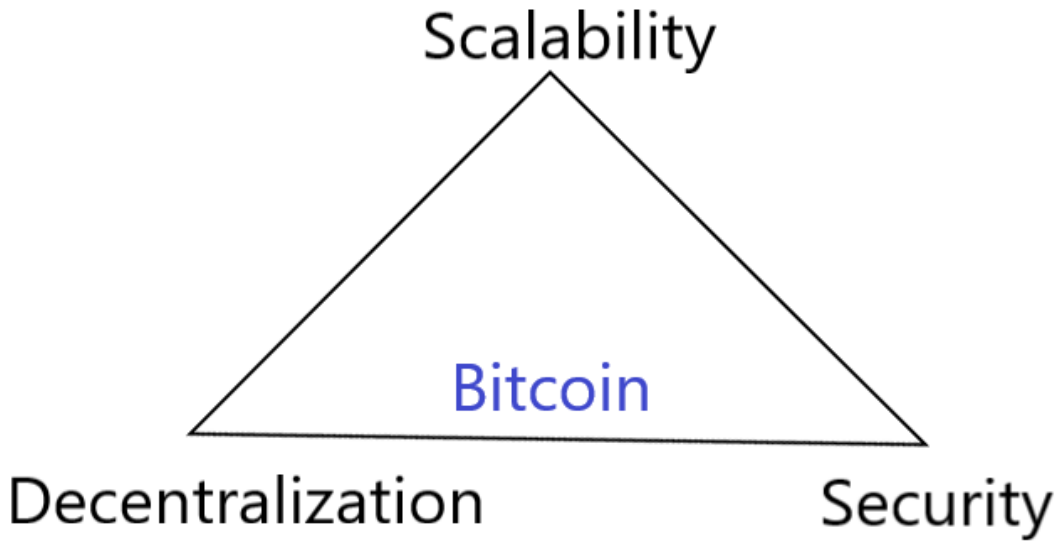




- Sharding
  - m shards: m blocks every 10 minutes
  - Linear increase in throughput with number of shards
- Bitcoin
  - 1 shard: 1 block every 10 minutes
  - Adding more resources does not increase throughput
  - Improve throughput for crypto adoption as transactional currency

Bitcoin's Throughput: ≈5 transactions per second
Visa's Throughput: ≈1700 transactions per second

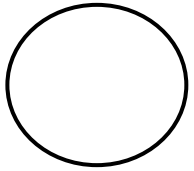
CAP Theorem / Blockchain Trilemma



- Pick one side of triangle
- Sharding lowers collective security of a system

Shard Size

Small shard

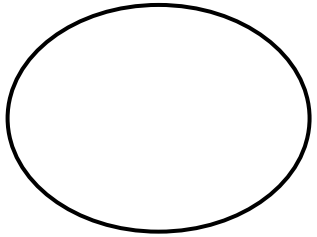


- Performance inside shard ↑

(small number of message exchanges)

- More shards = throughput ↑
- Security ↓

Large shard



- Performance inside shard ↓

(large number of message exchanges)

- Less shards = throughput ↓
- Security ↑

Aim : Middle ground between size (performance) & security

Network Model

Synchrony

- Message delivery bounded by fixed time ▲ (delta)
- If ▲=10 seconds, 6 blocks per minute
- Nonresponsive
- ▲ is a conservative value
- Wait for 10 seconds even if all messages delivered in 1 second

Partial Synchrony

- Message delivery occurs at actual network delay α (alpha)
- If α ≈ 1 second, 60 blocks per minute
- Responsive
- More practical

Aim:
Use Partial Synchrony

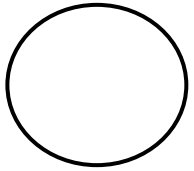
Fault/Adversarial Model

Traditional Systems

- Byzantine faults only
  - Arbitrary/Unpredictable behaviour
  - Attack Safety + Liveness
  - Most powerful type of fault
- Safety
  - Double-spend attack
  - Equivocation of values such as Merkle Root
  - Falsifying user balances
- Liveness
  - Preventing progress of system
  - Cutting off communication channels

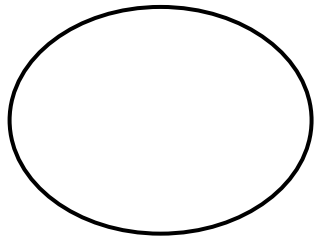
BFT Consensus Algorithms

Synchronous BFT



- $f < m/2$
- shard size, $m = 300$; $f \leq 149$
- More resilience = smaller shard size
  - Threshold = 10^{-10} : size ≈ 300
- Nonresponsive
- (RapidChain)

Partially Synchronous BFT



- $f < m/3$
- $m = 300$; $f \leq 99$
- Less resilience = larger shard size
  - Threshold = 10^{-10} : size ≈ 700 - 800
- Responsive
- (OmniLedger)

Aim :shard size comparable to RapidChain's but Responsive

Aim : Small Shard + Partial Synchrony (best of both worlds)

Fault/Adversarial Model

Traditional Systems

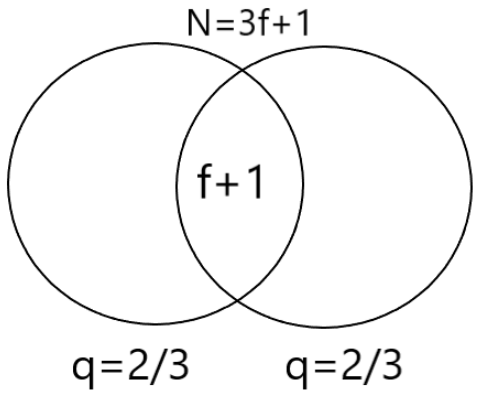
- Byzantine faults only
  - Arbitrary/Unpredictable behaviour
  - Attack Safety + Liveness
  - Most powerful type of fault
- Safety
  - Double-spend attack
  - Equivocation of values such as Merkle Root
  - Falsifying user balances
- Liveness
  - Preventing progress of system
  - Cutting off communication channels

FlexiShard

- Hybrid Fault Model
- Byzantine + abc faults
- abc : alive-but-corrupt faults
  - Attack Safety only
  - Collaborate to make progress if node cannot attack safety
  - More practical

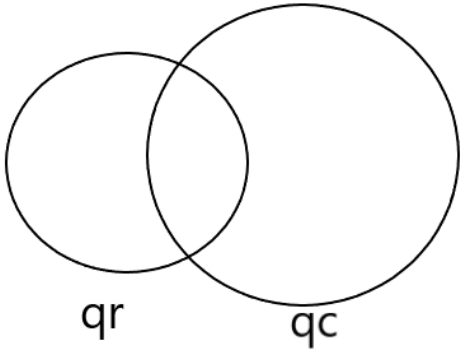
Partially Synchronous Consensus Algorithms

Classic BFT



- q = 2f+1 / 3f+1 ≈ 2/3 (optimal)
- PBFT : 2 rounds of voting
- 2f+1 identical votes per round
- Byzantine only
- F < 2q – 1
  - byzF < 2q – 1
  - abc = 0

Flexible BFT



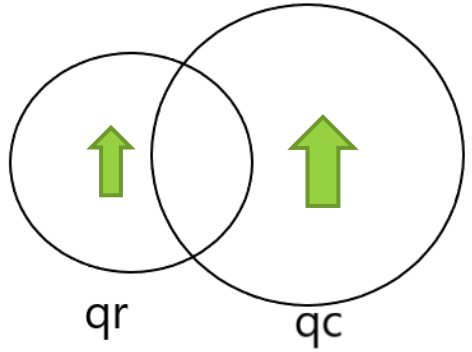
- Byzantine + abc faults
- qc ≥ qr
- F < qr + qc - 1
  - byzF ≤ 1 - qc
  - abc < F – byzF
  - abc < qr + 2qc -2

Partially Synchronous Consensus Algorithms

Flexible BFT Properties

- Byzantine + abc faults
- qc ≥ qr
- F < qr + qc - 1
  - byzF ≤ 1 - qc
  - abc < F – byzF

- Classic BFT
  - ⇧q: Lose Liveness (Quorum Availability)
  - ⇩q: Lose Safety (Quorum Intersection)
  - q = 2/3 is optimal



⇧ quorum size qr or qc or both :

Total number of faults ⇧

Byzantine faults ⇩

abc faults ⇧

Rate of abc ⇧ > Rate of Byz ⇩

Shard size, m = 300

Traditional	FlexiShard 1	FlexiShard 2	FlexiShard 3
Byzantine only	Byz+ abc	Byz+ abc	Byz+ abc
qr = 2/3	qr = 2/3	qr = 2/3	qr = 2/3
qc = 2/3	qc = 0.75	qc = 0.9	qc = 1
Total F ≤ 99 (33%)	Total F ≤ 125 (41%)	Total F ≤ 170 (56%)	Total F ≤ 199 (66%)
ByzF ≤ 99 (33%)	ByzF ≤ 75 (25%)	ByzF ≤ 30 (10%)	ByzF = 0 (0%)
abcF = 0 (0%)	abcF = 50 (16%)	abcF = 140 (46%)	abcF = 199 (66%)
Rigid/Conservative	Practical	Practical	Very Optimistic

Perfect Shards

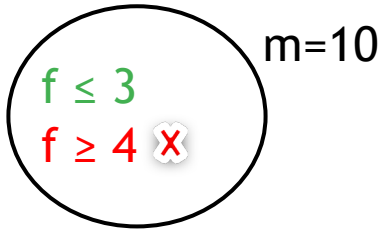
- Shards are assumed to be PERFECT
- Hence shards always produce correct results
- Need to form shards with NEGLIGIBLE (very low) failure probability

Calculation of Shard Failure probability (Traditional)

- BOUND THE FAILURE PROBABILITY
  - Actual failure probability never exceeds bounded probability (overestimation)

- m: shard size, F: total faults in system, N: total nodes in system, X: actual num of byz faults in shard

- Partial Sync, Byz only, f< m/3
- m=10, f ≤ 3 for safety or f ≥ 4 for failure



- Shard has failed when we have 4 faults OR 5 faults OR 6 faults OR … 10 faults.

- P_{Faulty} = P(f=4) + P(f=5) … + P(f=10)

- $P_{Faulty} = P\left(f \geq \left\lfloor \frac{m}{3} \right\rfloor\right) = \sum_{x=\lfloor \frac{m}{3} \rfloor}^{m} \frac{\binom{F}{x}\binom{N-F}{m-x}}{\binom{N}{m}}$

Calculation of Shard Failure probability (FlexiShard)

- BOUND THE FAILURE PROBABILITY
  - Actual failure probability never exceeds bounded probability (overestimation)

- m: shard size, F: total faults in system, N: total nodes in system, X: actual num of byz faults in shard

- Partial Sync, Byz + abc
- m=10, $f_{byz} \leq 4$ and $f_{abc} \leq 4$ for safety

- Shard has failed when either byz or abc threshold exceeded
  - $P_{Faulty} = P(f_{byz} \geq 5)$ OR $P(f_{abc} \geq 5)$

- Therefore, shard is safe when both type of faults are within their respective bounds
  - $P_{NonFaulty} = P(f_{byz} \leq 4)$ AND $P(f_{abc} \leq 4)$

Calculation of Shard Failure probability (FlexiShard)

- Shard is safe when both type of faults are within their respective bounds
  - $P_{NonFaulty} = P(f_{byz} \leq 4)$ AND $P(f_{abc} \leq 4)$
  - $P_{Faulty} = 1 - P_{NonFaulty}$

- Calculate all possibilities that a shard is safe:

$P_{NonFaulty} = P(f_{byz}=0 \text{ AND } f_{abc}=0) + P(f_{byz}=0 \text{ AND } f_{abc}=1) \ldots + P(f_{byz}=0 \text{ AND } f_{abc}=4)$

$+ P(f_{byz}=1 \text{ AND } f_{abc}=0) + P(f_{byz}=1 \text{ AND } f_{abc}=1) \ldots + P(f_{byz}=1 \text{ AND } f_{abc}=4)$

.

.

$+ P(f_{byz}=4 \text{ AND } f_{abc}=0) + P(f_{byz}=4 \text{ AND } f_{abc}=1) \ldots + P(f_{byz}=4 \text{ AND } f_{abc}=4)$

- $P_{Faulty} = 1 - P_{NonFaulty}$

Calculation of Shard Failure probability (FlexiShard)

- Hypergeometric Distribution Formula for mixed fault model: byzantine + abc faults

$$P_{NonFaultyFlexi} = P(Y \leq f_{abc}\ AND\ Z \leq f_{byz})$$

$$= \frac{\sum_{byz=0}^{fbyz} \binom{FB}{byz} \times \left(\sum_{abc=0}^{fabc} \binom{FA}{abc} \times \binom{N-F}{m-byz-abc} \right)}{\binom{N}{m}}$$

$$= \frac{\sum_{byz=0}^{\lfloor((1-qc)\times m)\rfloor} \binom{(\lfloor byzProp * F \rfloor)}{byz}}{\binom{N}{m}}$$

$$\times \left(\sum_{abc=0}^{\lfloor((qr+2qc-2)\times m)\rfloor} \binom{F-(\lfloor byzProp * F \rfloor)}{abc} \times \binom{N-F}{m-byz-abc} \right)$$

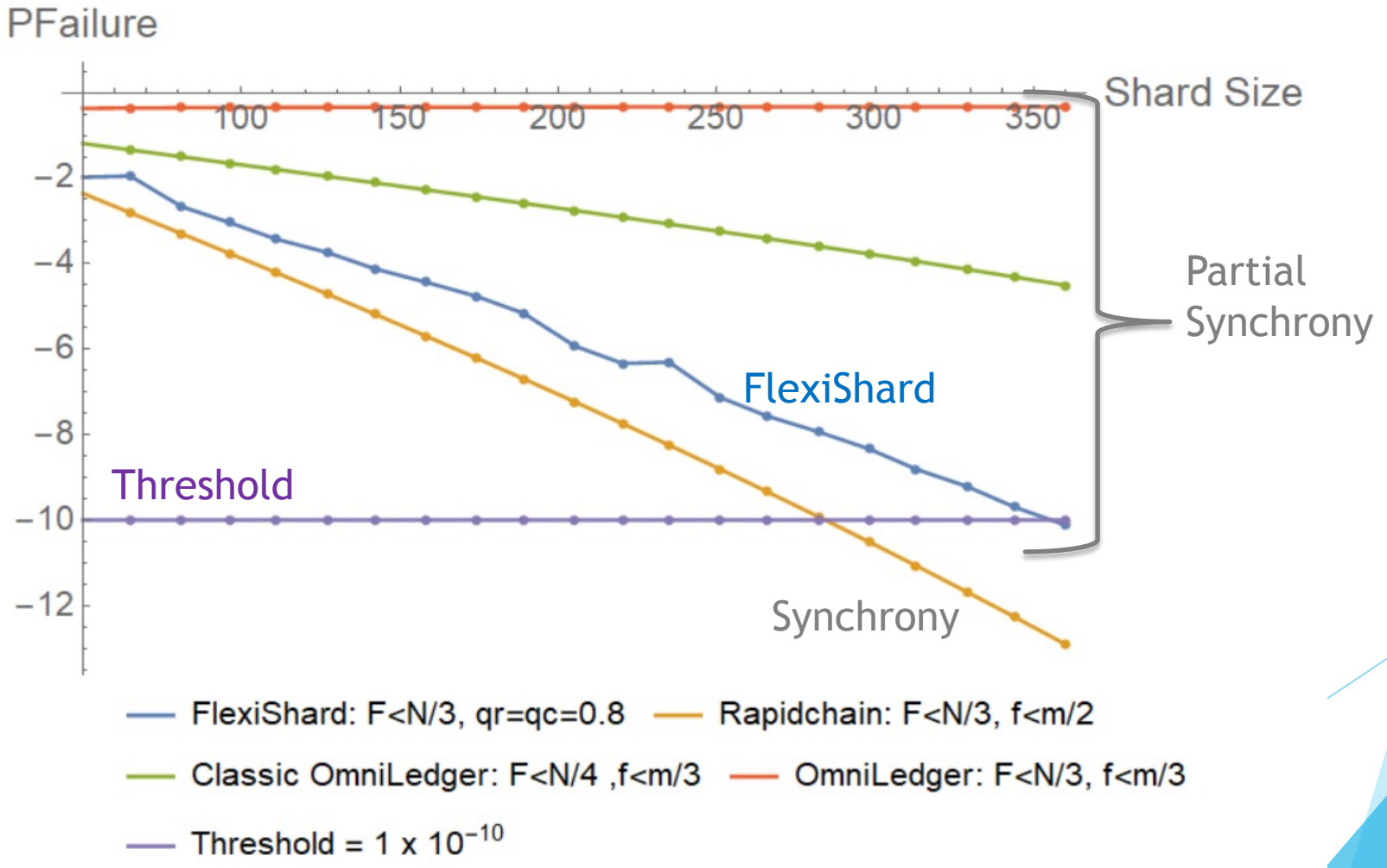
- $P_{FaultyFlexi} = 1 - PNon_{FaultyFlexi}$

FlexiShard's Schemes

- Many parameters can be varied
  - *F* , *f*, *m*, f_{abc}, f_{byz}, *byzProp* or *byz:abc split*, *qr*, *qc*
  - Vary parameters to adapt to an ever-changing security environment
- Traditional Systems: *F* and *f* only, and rigid values
- Different ways to shard a system
- Schemes in the paper describe ways you can shard a system
- Examples
  - How to obtain good quorum sizes
  - How to obtain good shard sizes for fixed quorum sizes
  - How to obtain an OPTIMAL shard size under different environments

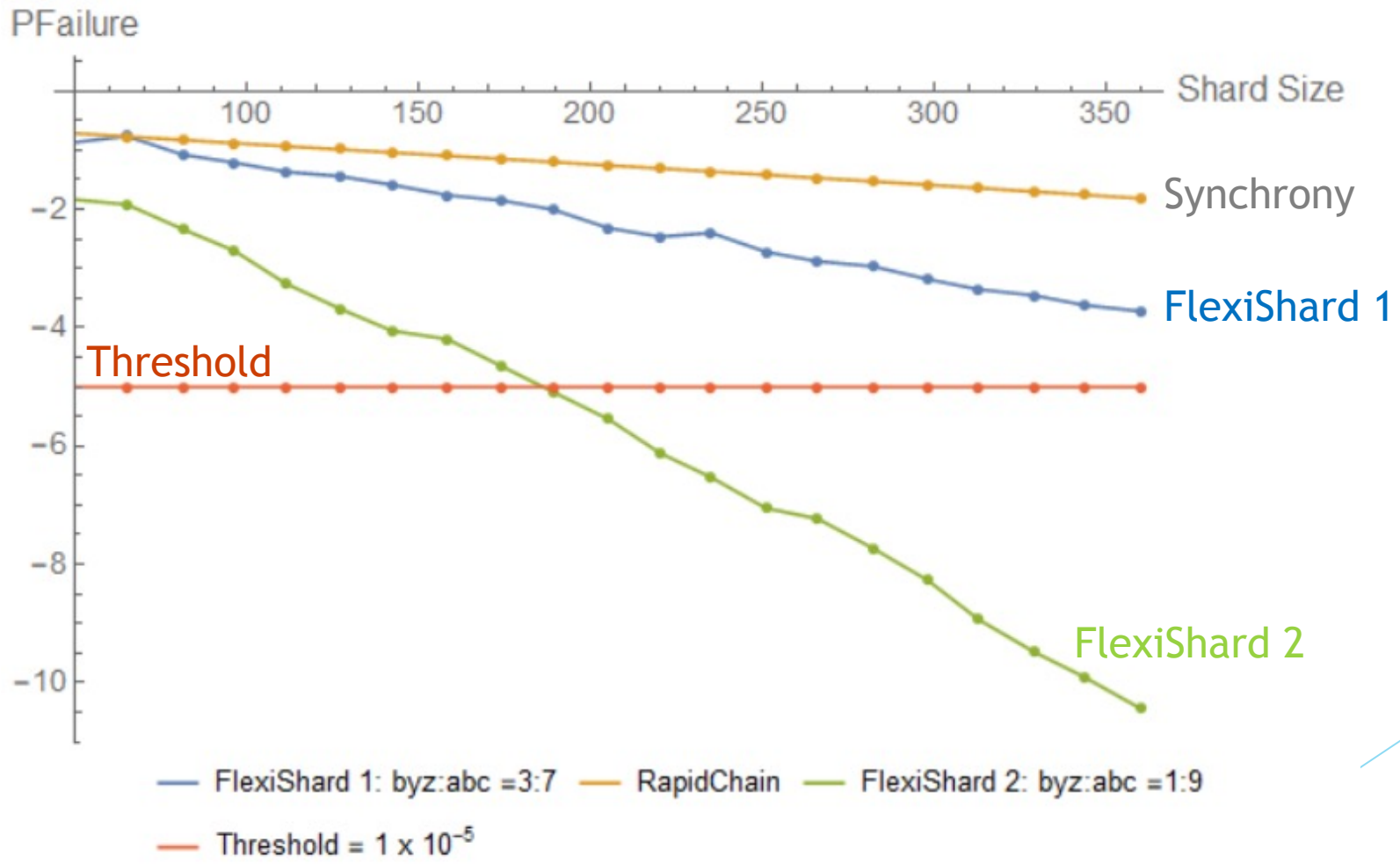
Results

Conventional bounds : F < 0.33 N



Results

Conventional bounds exceeded: F < 0.45 N



How is FlexiShard Useful?

Contemporary Sharding Systems

- Rigid parameters
- Difficult to adapt to a constantly-changing environment
- Synchrony
  - Intra-shard resiliency < 1/2
  - Total system resiliency < 1/3
  - Small shard
- Partial Synchrony
  - Intra-shard resiliency < 1/3
  - Total system resiliency < 1/4
  - Large shard size

FlexiShard

- Flexible parameters
- Can adapt accordingly & More Practical
- Intra-shard resiliency
  - Variable
  - Can exceed conventional bounds of 1/3 or 1/2
- Total system resiliency
  - Variable
  - Can exceed conventional bounds of 1/4 or 1/3
- Smaller shard size in Partial Synchrony
- Responsive
- Maximise performance potential
  - Small fault presence: extract more performance
  - Large fault presence: increase shard resiliency

Conclusion

- Hybrid Fault Model (More practical)
- Middle ground when total faults are within conventional bounds (1/3 or 1/4)
- Shards can also withstand more faults than conventional bounds in other cases

- Smaller shard sizes comparable to Synchrony; however in Partial Synchrony
- Responsive shards in Partial Synchrony (More Practical + Performance Gain)
- Middle ground between performance and security

THANK YOU

Appendix 1 – Components of a Sharding Blockchain

1. Node Selection (PoW, PoS)
2. Randomness generation (Secret Sharing, VFSS)
3. Node Assignment (Random Sampling + depends on consensus algorithm)
4. Intra-Shard Consensus (BFT – Sync or Partially Sync or Flexible)
5. Cross-shard Transaction Processing (Inter-Shard)
6. Shard Reconfiguration (Random, CuckooRule, Corruption Speed parameter)
7. Motivation Mechanism